

PREDICTION OF ESTIMATED ULTIMATE RECOVERY IN THE EAGLE FORD

A Thesis

by

MOHIT DHOLI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	David Schechter
Committee Members,	Bryan Maggard
	Yuefeng Sun
Head of Department,	A. Daniel Hill

May 2016

Major Subject: Petroleum Engineering

Copyright 2016 Mohit Dholi

ABSTRACT

The study presents prediction of Estimated Ultimate Recovery (EUR) for multi-stage hydraulically fractured horizontal wells producing primarily oil in the Eagle Ford. The EUR prediction models' comparison for the multi-stage hydraulically fractured horizontal wells in the Eagle Ford is made possible with the help of advances in neural networks. The monthly production and well data is collected for oil producing wells (1,134) drilled in 2010-11 in the Eagle Ford from Drilling Info Desktop. The models were trained using Multiple Linear Regression (MLR), Support Vector Machine (SVM) and Bayesian Regularized Neural Networks (BRNN). These models learn the relationship between well data and the EUR (estimated by decline curve analysis). Furthermore, these models were tested on the data not used in the training of the models. A model selection algorithm is formulated which produced a median absolute error of 22%.

The models were trained and tested using Eagle Ford shale oil production data but the methodology and code should be applicable to other resource plays as well.

This method could be useful for predicting the performance of various unconventional reservoirs for both oil and gas as a quick-look tool. As an advice for further work this tool can be used to prepare forecasts for unconventional gas reservoirs as well and combined with the oil forecasts to present a more holistic view.

DEDICATION

I would like to dedicate my work to my parents and brother for their love and trust, to my wife for her support and patience. Also, I dedicate this to all my friends and colleagues who helped me throughout my study and research. Last, but not the least, a special dedication to my late advisor, Dr. R.A. Wattenbarger for his supervision, guidance and support.

ACKNOWLEDGEMENTS

First of all, I thank God for His blessings that have steered me through this thesis. Also, I would like to thank my late advisor, Dr. R.A. Wattenbarger for his inspiration and help. In addition, I would like to express deep gratitude to the chair of my committee, Dr. David Schechter for his support and encouragement, especially after the sad demise of Dr. Wattenbarger. Special thanks to my committee members, Dr. Yuefeng Sun and Dr. Bryan Maggard for their advice and feedback.

I am grateful to my colleagues, Dr. Ahmad Alkoush, Mohammad Kanfar, Sinurat Pahala, Hussain AlDaif and Basel Alotaibi for their help during my research and study at Texas A&M University. They have made it an easier and fun activity.

I would like to take the opportunity to express my sincere thanks to Prof. John Jochen, Dr. David Schechter, Ms. Stacy Aschenbeck and Dr. Walter Ayers for employing me as a Graduate Assistant (Teaching), which gave me this opportunity by supporting me in my pursuit of the Master of Science degree.

The data for this study has been extracted from Drillinginfo and the coding is done in RStudio. I sincerely appreciate the efforts by Drillinginfo customer service to solve my queries during the study. Also, I thank the R core team to have produced and maintained such an optimized programming language for public use.

TABLE OF CONTENTS

	Page
ABSTRACT	ii
DEDICATION	iii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	vii
LIST OF TABLES	x
CHAPTER I INTRODUCTION	1
1.1 Prediction of Estimated Ultimate Recovery (EUR)	1
1.2 Objective and Motivation.....	3
1.3 Thesis Organization.....	3
CHAPTER II LITERATURE REVIEW.....	4
2.1 Arps' Decline Curve Analysis.....	4
2.2 R	6
2.3 Multiple Linear Regression	8
2.4 Support Vector Machine	10
2.5 Bayesian Regularized Neural Networks	12
CHAPTER III DATA SOURCES.....	14
3.1 Drillinginfo (Website and DI Desktop).....	14
3.2 Evaluation of Estimated Ultimate Recovery (EUR)	17
CHAPTER IV EXPLORATORY DATA ANALYSIS	24
4.1 Single Variable Analysis	24
4.2 Outlier Analysis.....	37
CHAPTER V MODELING AND RESULTS	41
5.1 Methodology and Results from Modeling	41

5.2 Model Selection Algorithm.....	48
CHAPTER VI DISCUSSION AND CONCLUSIONS	51
REFERENCES	55
APPENDIX A FIGURES AND CODE	57
APPENDIX B OUTLIER ANALYSIS.....	83

LIST OF FIGURES

	Page
Figure 1. Arps’ three types of decline curves on a semi-log plot (Arps 1945).	5
Figure 2. An example of simple linear regression showing the scatter plot and the line of least squares estimates. The data set “iris” is provided by default in the R source code and has been reproduced here.	9
Figure 3. Demonstration of hyperplanes in the high dimensional feature space. The hyperplane H2 does not even separate the classes. Between hyperplane H1&H3,H3 separates the nearest point of the two classes with maximum distance hence creating largest separation.	11
Figure 4. Illustration of Single layer feed forward neural network. The x values are the inputs where as y is the predicted value. (Perez-Rodriguez et al. (2013))	13
Figure 5. The map of Texas showing the location of active oil wells drilled in the Eagle Ford for the year 2010-11 (DID).	15
Figure 6. Map showing the counties of Texas to which the wells belong. (NTS)	15
Figure 7. A chart showing the number of active horizontal wells drilled in each county of Eagle Ford, Texas in the year 2010-11 (DID).	16
Figure 8. A chart showing the number of active horizontal wells drilled for major operators of Eagle Ford, Texas in the year 2010-11 (DID). The category “Others” include all the operators having less than 10 wells drilled.	17
Figure 9. A typical decline curve ($D_i = 1.335$ /year, $b = 0.355$) for the EUR calculation of oil wells drilled in 2010-11 in the Eagle Ford. The forecast lines are extended for illustration purpose as the well’s economic limit is 150 STB/month.	18
Figure 10. The Histogram chart of latest liquid rate is shown for all the oil wells drilled in 2010-11. The economic limit of 150 STB/month was chosen as 90% of the last liquid rates are above it.	19
Figure 11. A typical decline curve ($D_i = 1.3$ /year, $b = 0.18$) for an oil well in the Eagle Ford showing no production from the months 14-23. The EUR is	

calculated for abandonment rate of 150 STB/month. The forecast lines are extended just for illustration purpose.	20
Figure 12. Chart showing box-plots of EUR determined for the oil wells drilled in 2010-11 of the 17 counties of Eagle Ford, Texas.....	21
Figure 13. Chart showing box-plots of EUR determined for the oil wells drilled in 2010-11 for the major operators of Eagle Ford, Texas.....	22
Figure 14. An illustration of how to interpret a boxplot.	23
Figure 15. A screen shot of the CSV data sheet showing a few data rows. The input column of GOR and First month's water production each show missing values which are then removed from the analysis.	26
Figure 16. A scatter plot of EUR vs. the peak monthly oil rate for the well with a linear regression line. All the data points are plotted here minus the missing values.	26
Figure 17. A scatter plot of EUR vs. the liquid gravity for the well with a linear regression line. All the data points are plotted here minus the missing values.	27
Figure 18. A scatter plot of EUR vs. the Total Depth for the well with a linear regression line. All the data points are plotted here minus the missing values.	28
Figure 19. A scatter plot of EUR vs. the First month's water production for the well with a linear regression line.	29
Figure 20. A scatter plot expanding Fig. 19 for the convenience of the reader.	30
Figure 21. Scatter plots of EUR vs. First month's water production for Atascosa, Wilson, LaSalle and Gonzales with a linear regression line. The plots also report the correlation coefficient.	31
Figure 22. Scatter plots of EUR vs. First month's water production for major operators with a linear regression line. The plots also report the correlation coefficient.	32
Figure 23. A scatter plot of EUR vs. the completed interval length (Lateral Length) for the wells with a linear regression line.	33
Figure 24. Scatter plots of EUR vs. Lateral length for Atascosa, Dewitt, Gonzales, Karnes County and major operators with a linear regression line. The plots also report the correlation coefficient.	34

Figure 25. Scatter plots of EUR vs. Gas-Oil ratio for the wells. The top Plot shows GOR range till 20,000 Mscf/STB whereas below one displays a deeper look.	35
Figure 26. Scatter plots of EUR vs. Gas-Oil ratio for Atascosa, Dimmit, Live Oak, Gonzales County and major operators with a linear regression line. The plots also report the correlation coefficient.	36
Figure 27. Boxplots of various input and output variables before and after outlier analysis.	39
Figure 28. Boxplots showing outliers for “Total Depth” and “Liquid Gravity” input variables.	40
Figure 29. A plot showing how the optimum number of clusters are determined.	42
Figure 30. A 3D plot showing 3 input variables clustered using k-means algorithm. The clustering is done on all the 5 inputs.	43
Figure 31. A flowchart depicting the progress of the modeling process.	44
Figure 32. Comparison of median percentage absolute errors for the whole data and the County clusters. LM stands for Linear Model (MLR).	46
Figure 33. Comparison of median percentage absolute errors for the whole data and the algorithmic clusters. LM stands for Linear Model (MLR).	47
Figure 34. Flow chart depicting the process of model selection.	48
Figure 35. A histogram of Absolute percentage errors for the testing data obtained from the model selection algorithm. The median is shown with the use of white dashed line at 22.00%	49
Figure 36. An illustrative plot of decline curve showing the range of median percentage error in the prediction of EUR using the model selection algorithm.	50

LIST OF TABLES

	Page
Table 1-Arps equations for rate and cumulative determination (Arps (1945)).....	5
Table 2. List of the external packages and their title used in this work other than the standard ones provided with the source code.	7
Table 3. List of Input and Output variables considered in this study.....	25
Table 4. List of final input and output variables for analysis.....	37
Table 5. List of data sets used in modeling.	42

CHAPTER I

INTRODUCTION

1.1 Prediction of Estimated Ultimate Recovery (EUR)

The latest downturn in oil prices has already led to more than 150,000 lay-offs worldwide (Jones (2015)) and analysts are anticipating 2016 to be another difficult year. Oil and gas companies are postponing their plans to invest in capital intensive projects and concentrate on their strengths to keep the company afloat. It is in these times, we truly understand the reach of technology and science. It is imperative to identify sweet spots in resource plays to keep on producing even in low price scenarios. Dry or low EUR wells would be a blow to any company's finances.

This method of predicting EUR using the well data could be a useful tool for producers to concentrate in the best region of their assets. The shale plays in the US possess complex heterogeneity which are difficult to disentangle. Understanding the spatial distributions and factors affecting these heterogeneities are very important. They also affect the cumulative production which is the ultimate product. The Estimated Ultimate Recovery is the sum of resources that could be produced from a well and the cumulative production till date. As more and more production data is recorded, accuracy increases. EUR is the key economic figure in deciding whether or not to invest millions of dollar to drill a well. There are several deterministic and probabilistic approaches to obtain EUR. All of them would need several months of production data. By predicting the

EUR using well data, we partially eliminate the need of production data to quantify EUR within a certain accuracy.

Machine learning approaches have been applied in a wide variety of fields to resolve complex problems such as classification, feature identification, investigation and optimization. Here, neural networks and machine learning are employed to produce models of EUR prediction.

Neural networks could be particularly useful when the data is noisy and when unknown non-linearity exists between independent and dependent variables (Bhatt (2002)). Neural networks are most likely to be better than others with the following conditions (Masters (1993)):

- a. The data on which the output is based is subject to probably large errors.
- b. The pattern important to the realization of output are subtle and obscured. Neural networks has the advantage of discovering hidden patterns within the data, not perceptible to the human brain or standard statistical methods.
- c. The data is scattered everywhere and chaotic.
- d. The data shows significant non-linear distributions.

The geological and completions data in the petroleum industry follows the fashion listed above. In this case, it makes sense to employ neural networks.

Thus, this study explores the performance of neural networks on real production and well data. It shows the application and comparison of SVM, BRNN and MLR in predicting EUR in Eagle Ford.

1.2 Objective and Motivation

To be able to predict EUR with the help of a model selection approach combining results from MLR, SVM and BRNN. The objective is to provide another tool to the industry that could be employed in effective decision making before drilling a well.

1.3 Thesis Organization

The organization of these chapter is as follows:

Chapter I is an introduction to the subject of this research, its motivation and objectives.

Chapter II is a literature review of Arps' Decline curve analysis, various algorithms utilized in the study, R programming and Eagle Ford shale oil field.

Chapter III explains the data sources as well as the variety of data extracted from these sources.

Chapter IV explains the methodology adopted for exploratory data analysis.

Chapter V investigates the results and model selection algorithm.

Chapter VI discusses the conclusions.

CHAPTER II

LITERATURE REVIEW

Since the advent of unconventional resources, several authors have published their studies on EUR determination, errors and appropriate methods to follow (Swindell (2012), Crnkovic-Friis and Erlandson (2015), Valko and Lee (2010), Chen et al. (2015) and Gao and Gao (2013)). Crnkovic-Friis and Erlandson (2015) present a geology driven approach to predict EUR using deep learning algorithms. Most of the other methods and work requires the use of production data to obtain EUR. The machine learning approach that is proposed in this study does obtain EUR but in the absence of production data. The method uses public well data to predict EUR using MLR, SVM and BRNN. In the end, a program is coded to automate the whole process of model selection and prediction.

2.1 Arps' Decline Curve Analysis

Arps' describes three types of decline curves during boundary dominated flow (BDF) based on the loss-ratio method: Exponential, Hyperbolic and Harmonic. Depending on whether the value of loss-ratio to be following a constant, arithmetic or geometric series, the decline curve falls in on or the other category. It is a graphical method to extrapolate the production in BDF to the economic limit and calculate the cumulative production. It is a plot of $\log q$ vs t (Fig. 1). Arps' defines the mathematical equations and the derivatives to obtain the parameters. The equations are described in Table 1.

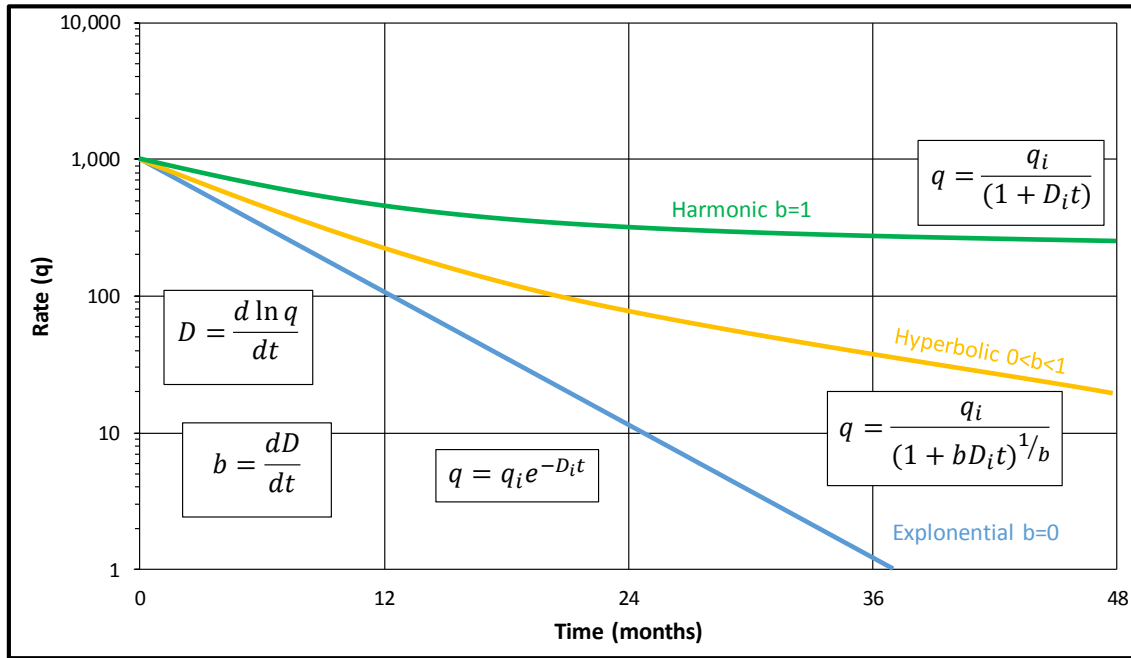


Figure 1. Arps' three types of decline curves on a semi-log plot (Arps 1945).

Table 1-Arps equations for rate and cumulative determination (Arps (1945))

Exponential ($b = 0$)	Hyperbolic ($0 < b < 1$)	Harmonic ($b > 1$)
$D = \text{constant}$	$D = \text{changing}$	$D = \text{changing}$
$q = q_0 e^{-tD_i}$	$q = q_0 (1 + b D_i t)^{-\frac{1}{b}}$	$q = q_0 (1 + b D_i t)^{-1}$
$Q = \frac{(q_0 - q)}{D_i}$	$Q = \frac{q_0^b}{D_i(1-b)} (q_0^{1-b} - q^{1-b})$	$Q = \frac{q_0}{D_i} (\log q_0 - \log q)$

2.2 R

R is a well-known and unified suite of free software facilities and computer programming language for statistical computations and graphical display. As a data analysis software its strength lies in an efficient data handling and storage, a batch of operations on matrices and data tables, integrated data analysis tools, well-designed publication-quality graphics and plots, a well-developed and open source programming language (R Core Team (2015)).

The R source code comes with default functions which are called *packages*. This is done for increasing performance as it saves memory and to prevent package developers' name clashes (R Core Team (2015)). These extra packages contain specialized functions to aid the programmer and are available through CRAN site (Comprehensive R Archive Network, <http://cran.r-project.org>).

This study uses many external packages other than the standard packages given with the R software for which a list is provided in Table 2. These libraries should be installed and uploaded to R for the code to execute.

Table 2. List of the external packages and their titles used in this work other than the standard ones provided with the R source code.

S. No.	CRAN	Title
	Package	
1	brnn	Bayesian Regularization for Feed-Forward Neural Networks
2	car	Companion to Applied Regression
3	caret	Classification and Regression Training
4	e1071	Misc. Functions of the Department of Statistics, Probability Theory Group, TU Wien
5	FSelector	Selecting attributes
6	gdata	Various R Programming Tools for Data Manipulation
7	ggplot2	An Implementation of the Grammar of Graphics
8	kernlab	Kernel-Based Machine Learning Lab
9	lattice	Trellis Graphics for R
10	latticeExtra	Extra Graphical Utilities Based on Lattice
11	rgl	3D Visualization Using OpenGL
12	rJava	Low-Level R to Java Interface
13	rpart	Recursive Partitioning and Regression Trees

2.3 Multiple Linear Regression

Regression analysis is the examination of any useful relationship among two or more variables (Sheather (2008)). Problems may involve investigating correlation, if any, between two variables or three or more variables. They are called simple linear regression or multiple linear regression respectively (Sheather (2008)).

The linear regression model can be written in the matrix form as

$$Y = X\beta + \epsilon \dots\dots\dots (1)$$

where,

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, X = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix}, \beta = \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_p \end{bmatrix}, \epsilon = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

$y_1, y_2 \dots y_n$: *Dependent variables*

$x_{11}, x_{12} \dots x_{np}$: *Independent variables*

β : *Regression coefficients*

ϵ : *Error*

The least squares estimates of $\beta_0, \beta_1, \beta_2, \dots, \beta_p$ are $\widehat{\beta}_0, \widehat{\beta}_1, \widehat{\beta}_2, \dots, \widehat{\beta}_p$ and are determined by minimizing the sum of the squared residuals (Sheather (2008)),

$$RSS = \sum_{i=1}^n \widehat{e}_i^2 = \sum_{i=1}^n (y_i - \widehat{y}_i)^2 = \sum_{i=1}^n (y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_{1i} - \widehat{\beta}_2 x_{2i} - \cdots - \widehat{\beta}_p x_{pi})^2$$

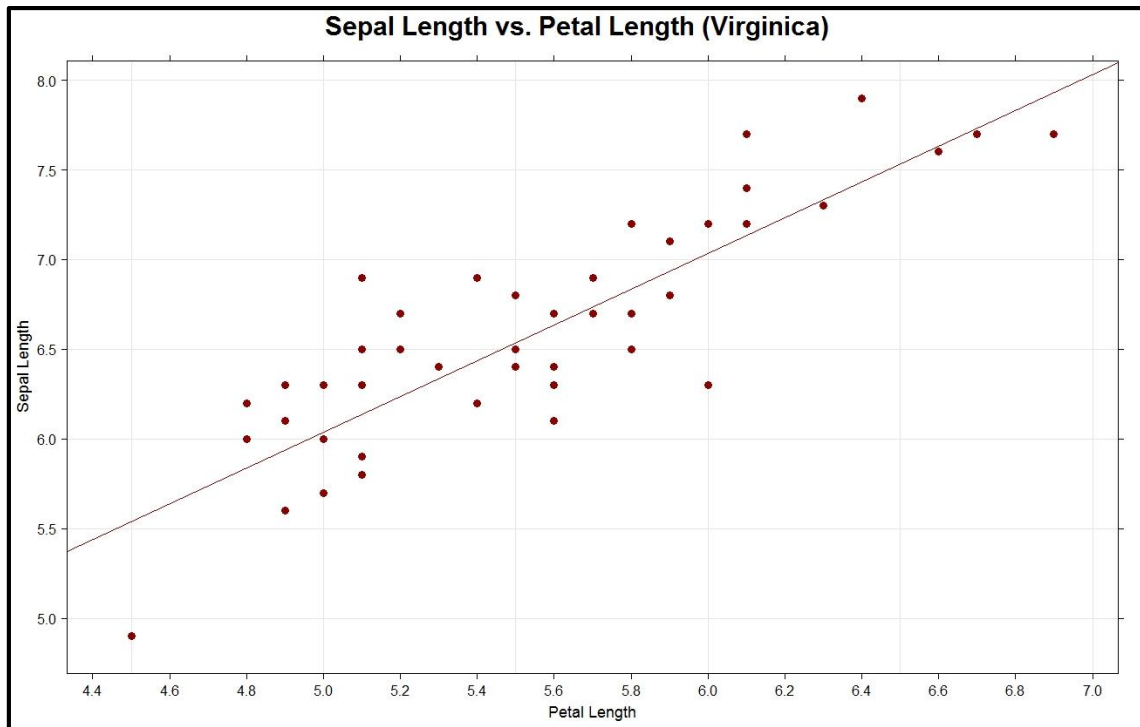


Figure 2. An example of simple linear regression showing the scatter plot and the line of least squares estimates. The data set “iris” is provided by default in the R source code and has been reproduced here.

The scatter plot in the Fig. 2 shows the linear regression between the dependent variable Sepal Length and the independent variable Petal Length. The “iris” data set has been used to prepare this chart which comes by default with the R software.

2.4 Support Vector Machine

Support Vector Machine (SVM) is a powerful machine learning approach to the classification and regression analysis (Cortes and Vapnik 1995). It is a supervised learning approach to the two-group classification problems. It is considered to be easier to implement than Neural Networks (Hsu et al. (2010)). Moreover, statisticians, programmers and other users are able to understand the processing during the implementation better than for Neural Networks which are famously opaque.

The data set is divided into two groups of training and testing data sets. The training data set consists of m points (x_i, y_i) where $i = 1, 2, \dots, m$. Every x_i is a feature vector consisting of n dimensions that defines the data point while every y_i has target binary values (± 1) (Cortes and Vapnik (1995)). The SVM model maps the input data non-linearly to a very high dimensional feature space separating the observed data into two-groups with one or more linear hyperplanes. The hyperplane which shows the maximum distance from the nearest training data point is then selected as the hyperplane. The function of this linear hyperplane is the model which is denoted by the support vectors. It predicts the value of y_i not only for the training but also for the testing data set. Cortes and Vapnik (1995) compare the performance of SVM to other algorithms and show that SVM performs better.

A demonstration (Fig. 3) of the above explanation is shown with 3 linear hyperplanes separating the classes. The hyperplane producing largest separation is H_3 .

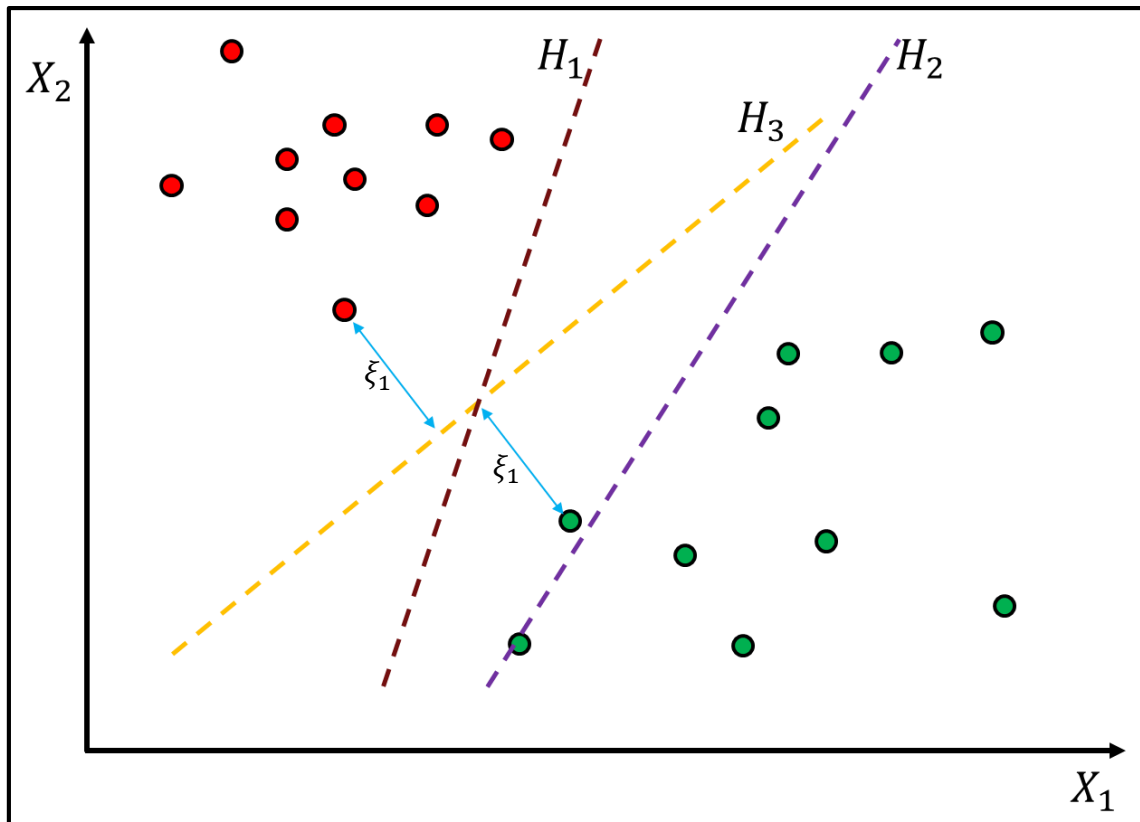


Figure 3. Demonstration of hyperplanes in the high dimensional feature space. The hyperplane H_2 does not even separate the classes. Between hyperplane H_1 & H_3 , H_3 separates the nearest point of the two classes with maximum distance hence creating largest separation.

2.5 Bayesian Regularized Neural Networks

The concept of Neural Networks (NN) had been developed as a functioning alter ego of the human brain which consists of billions of neurons helping it to perform its duties. Typically, the brain is capable of thinking, controlling the body movements, planning, optical visualization, responding to languages etc. It is a very complex process and involves non-linear and simultaneous computations. These neurons help the brain and spinal cord to regulate bodily functions by storing and transmitting gained information through electrical and chemical processes.

NN emulates the network of neurons in human brain and spinal cord by learning while processing the functions and distributing the acquired information through inter-neuron networks. Thus a neuron becomes the simplest processing unit of NN.

NN, as mathematical models, have been used for data fitting and predictions in many fields of research (Perez-Rodriguez et al. (2013)). The authors also found that the prediction capability of BRNN is better than linear models. Perez-Rodriguez et al. (2013) also developed the R package that this study uses for preparing BRNN models.

A basic NN is the Single Hidden Layer Feed Forward Neural Network (SLNN) (Fig. 4) which non-linearly transforms the inputs in the hidden layer and then combine them linearly to obtain the predictions (Perez-Rodriguez et al. (2013)).

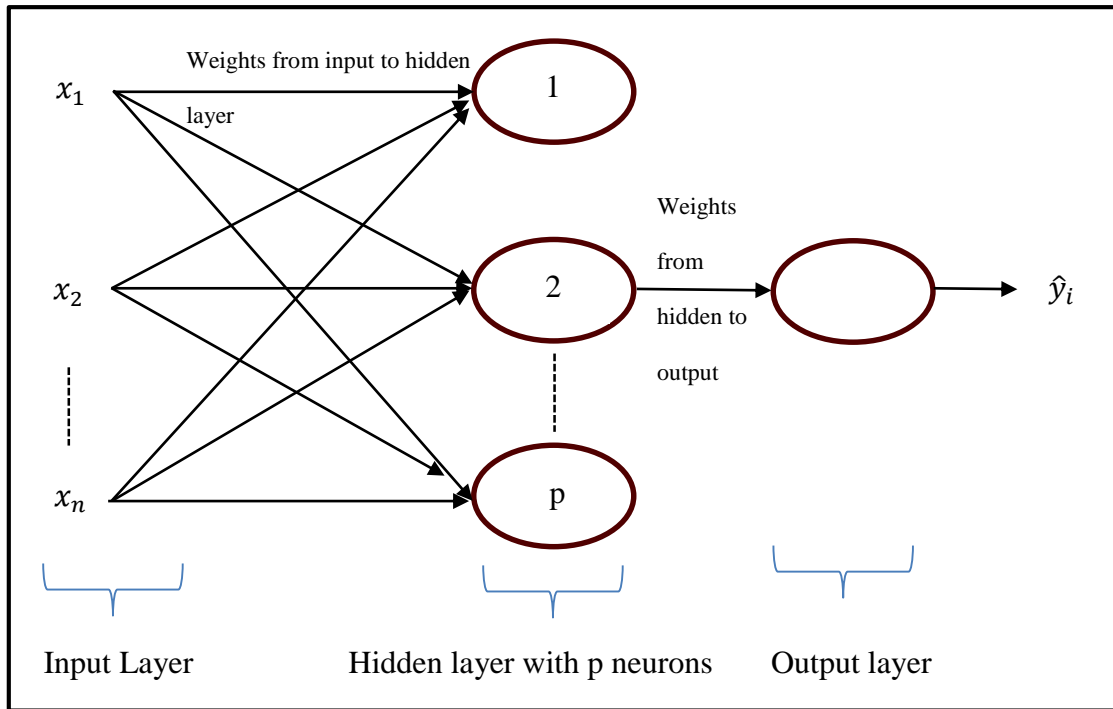


Figure 4. Illustration of Single layer feed forward neural network. The x values are the inputs where as \hat{y} is the predicted value. (Perez-Rodriguez et al. (2013))

NN is a very flexible modeling mechanism which when combines with higher number of inputs and neurons may result in overfitting of the data. Bayesian method corrects this issue by penalizing the estimation (Perez-Rodriguez et al. (2013)).

CHAPTER III

DATA SOURCES

3.1 Drillinginfo (Website and DI Desktop)

Drillinginfo (DI) is a comprehensive information resource web-tool for oil and gas industry in the U.S. DI provides the most up-to-date land & leasing, well, and regulatory data through a powerful, easy-to-use search application. It delivers ready-to-serve results in user-friendly formats that make it easy to digest and distribute. Drillinginfo tools combined with the huge amount of data help operators to make decisions with effective reasoning.

Drillinginfo Desktop (DID), the desktop application of Drillinginfo, was the main source of production and other well data for this study. The data was filtered to the expectation of the user and even mapped to visualize it. The study required significant amount of production data for the Eagle Ford oil wells to calculate Estimated Ultimate Recovery (EUR). Hence, only 2010-11 drilled wells were queried which are still active and producing.

Fig. 5 shows the horizontal wells in the map selector from DI Desktop. The wells are color coded for the year they are drilled in. Fig. 6 highlights the counties to which the data belongs. A total of 1,134 horizontal active oil wells were found to be drilled in 2010-11 in the Eagle Ford.

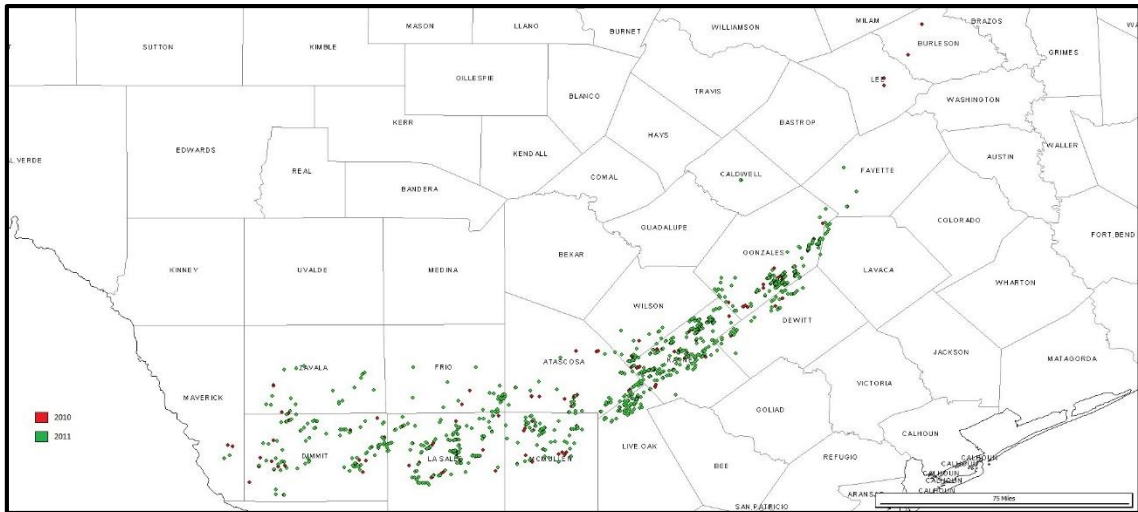


Figure 5. The map of Texas showing the location of active oil wells drilled in the Eagle Ford for the year 2010-11 (DID).

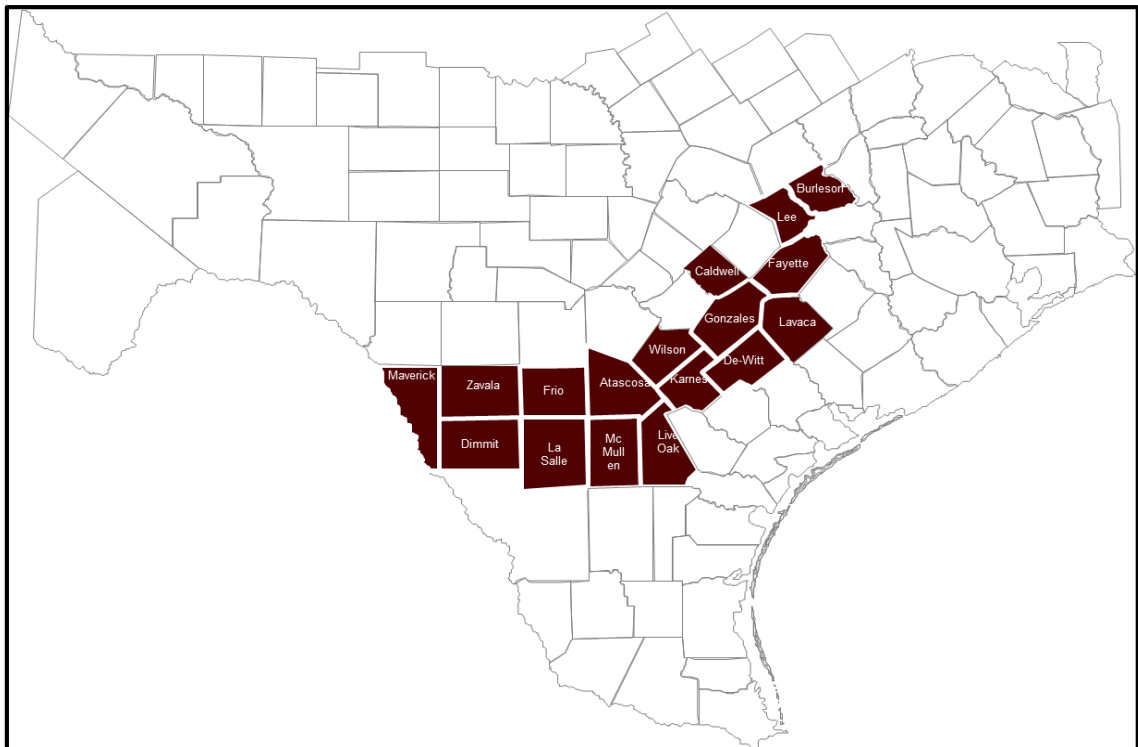


Figure 6. Map showing the counties of Texas to which the wells belong. (NTS)

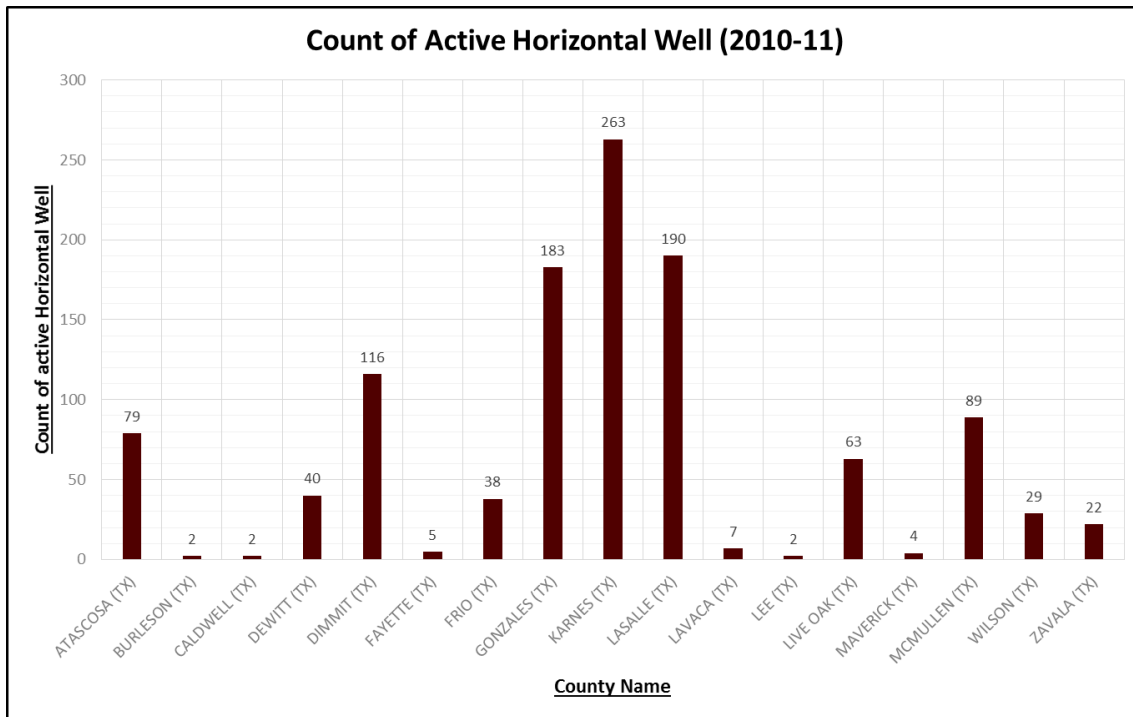


Figure 7. A chart showing the number of active horizontal wells drilled in each county of Eagle Ford, Texas in the year 2010-11 (DID).

The Fig. 7 charts the horizontal wells categorizing them by the county to which they belong. It is to be noted that 2010-11 were the early days of exploring oil in Eagle Ford shale and operators were still searching for their sweet spots. The chart shows that more than half of the wells exist in the top three counties i.e. Karnes, LaSalle and Gonzales. Counties like Burleson, Caldwell Lee, Fayette and Maverick have very few wells drilled.

The chart shown in Fig. 8 depicts the number of horizontal wells drilled by respective operators. The chart shows all the operators which have drilled at least 10 or more than 10 oil wells in 2010-11. There are many operators which have less than 10 oil

wells and are clubbed together in “others” category. It shows that the top five operators contribute more than half of the wells. It would not be surprising to note that most of EOG Resources wells belong to Karnes, LaSalle and Gonzales counties.

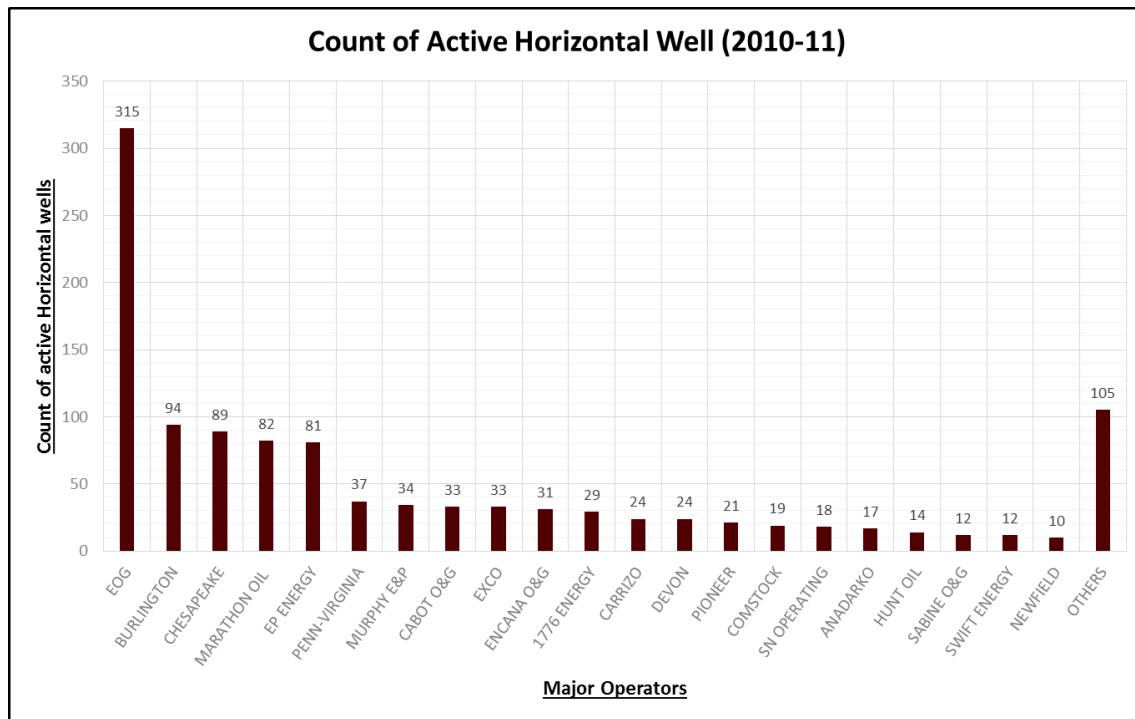


Figure 8. A chart showing the number of active horizontal wells drilled for major operators of Eagle Ford, Texas in the year 2010-11 (DID). The category “Others” include all the operators having less than 10 wells drilled.

3.2 Evaluation of Estimated Ultimate Recovery (EUR)

DID reports EUR for wells chosen according to the production data which has been reported till date. However, EUR calculations aren’t flexible enough. The primary product of the well, oil or gas, is decided on the last 12 month cumulative Gas-Oil ratio as well as the economic limit is pre-defined. Keeping in mind all these constraints, the EUR

calculation for all the 1,134 wells were done using Arps' decline curve analysis. The economic limit for this particular analysis was chosen to be 150 STB/month. The choice of the rate was done after a careful study of the last month liquid volume production. A histogram of last month's liquid rates (Fig. 10) shows that 90% of these rates lie above 150 STB/month and hence would engulf most of the wells.

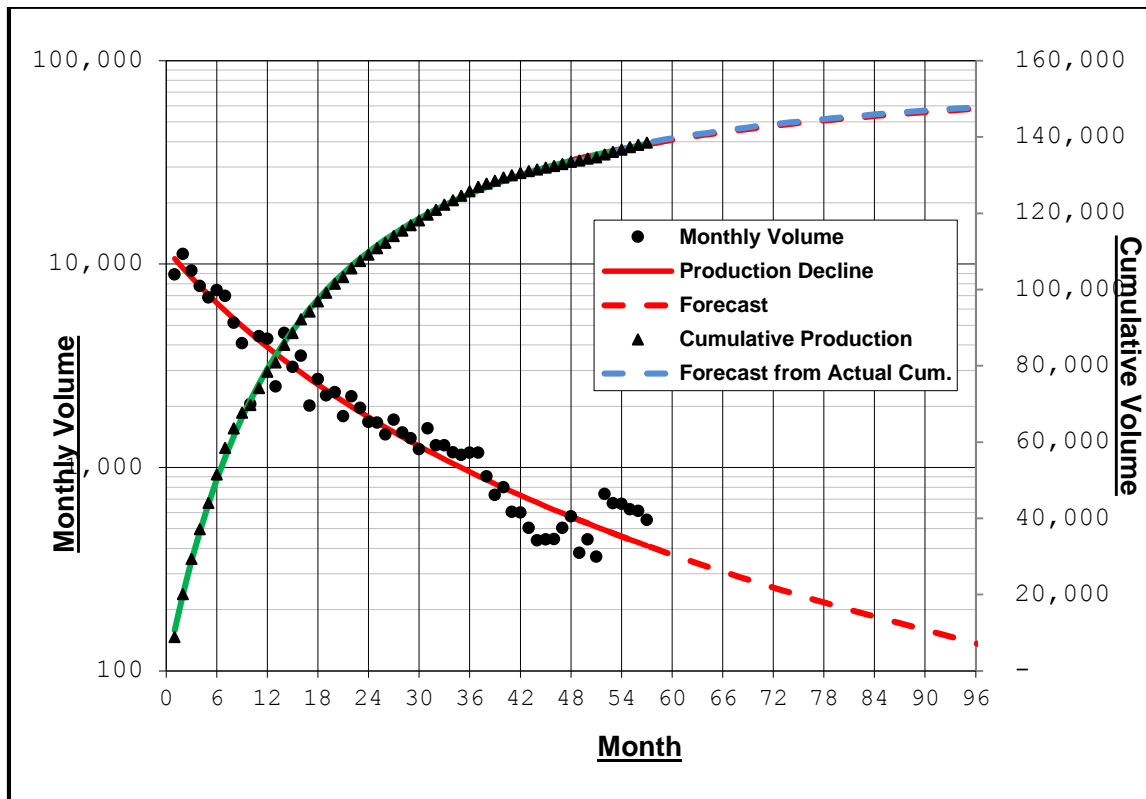


Figure 9. A typical decline curve ($D_i = 1.335$ /year, $b = 0.355$) for the EUR calculation of oil wells drilled in 2010-11 in the Eagle Ford. The forecast lines are extended for illustration purpose as the well's economic limit is 150 STB/month.

The production data obtained from DID was not always ideal. In more than a few cases, production data was missing for months in between for a well. This turned out to

be a major hurdle in the EUR calculation. The production data would start from its first production month, after continuing to produce for much time it will miss data for 2-6 months (could be more or less). After enquiring with the DID technical team, I was notified that the missing months have no production. In turn, the well was not producing for these months in between and then started the production again. This could be due to recompletion since in almost all cases the restarted production rate was high.

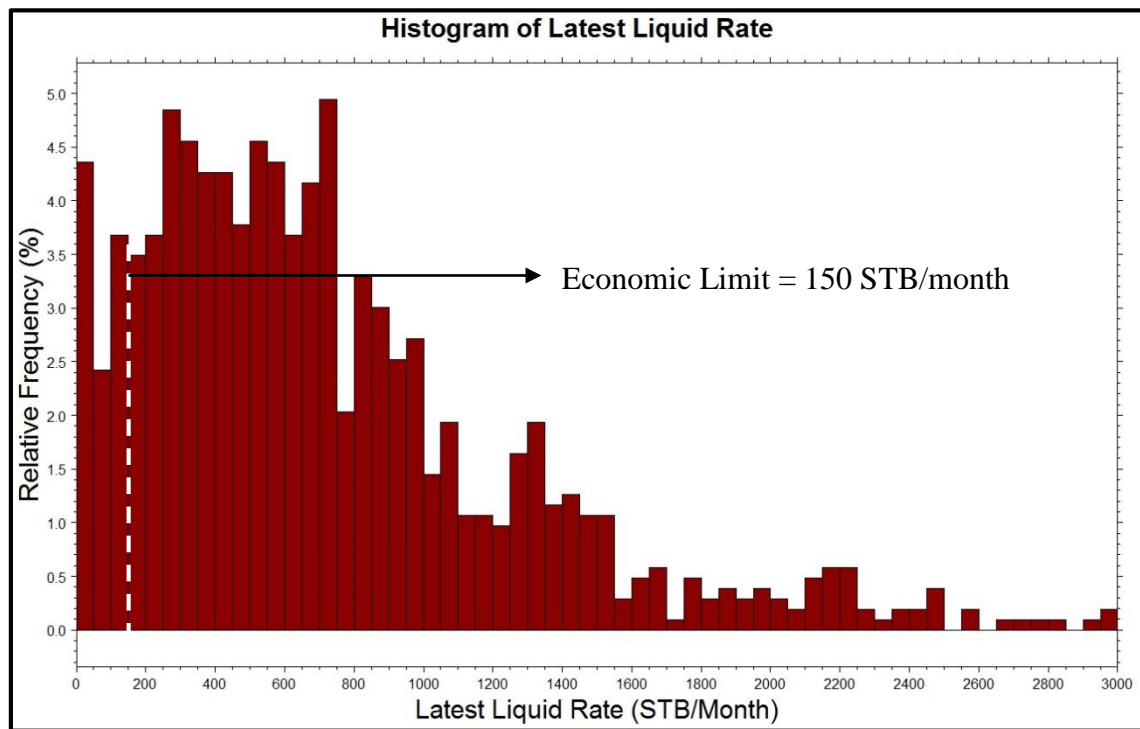


Figure 10. The Histogram chart of latest liquid rate is shown for all the oil wells drilled in 2010-11. The economic limit of 150 STB/month was chosen as 90% of the last liquid rates are above it.

A Visual Basic code was written to automate the process of EUR determination using the monthly production data of each well from the Eagle Ford. A typical decline

curve for the active oil well drilled in the Eagle Ford is shown in Fig. 9, where the well had been producing every month for its life.

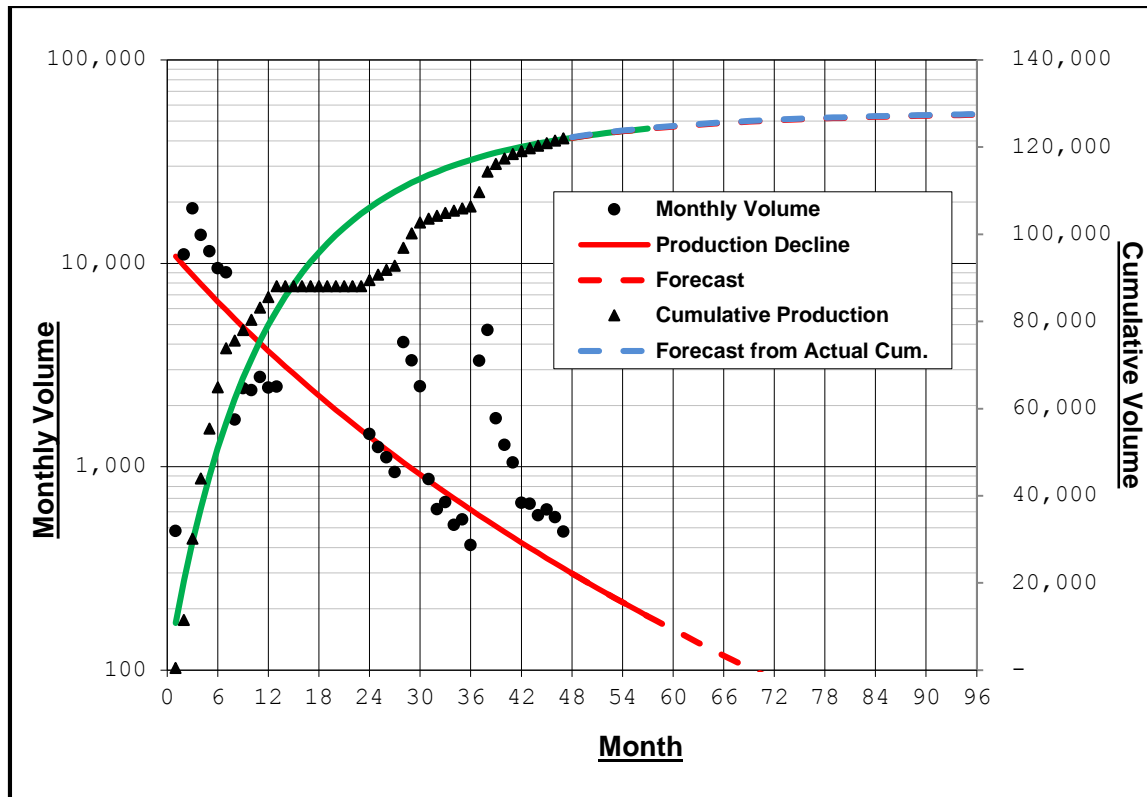


Figure 11. A typical decline curve ($D_i = 1.3/\text{year}$, $b = 0.18$) for an oil well in the Eagle Ford showing no production from the months 14-23. The EUR is calculated for abandonment rate of 150 STB/month. The forecast lines are extended just for illustration purpose.

In spite of automating the process of determining EUR to some extent, it took a lot of time and due diligence to complete the 1,134 wells' calculation. It is imperative to replace the missing data with zeroes for the corresponding months and at the end, check the EUR. In this prediction of EUR study, ensuring that the EUR calculations are correct is necessary for each data point as this directly affects the error and the final result.

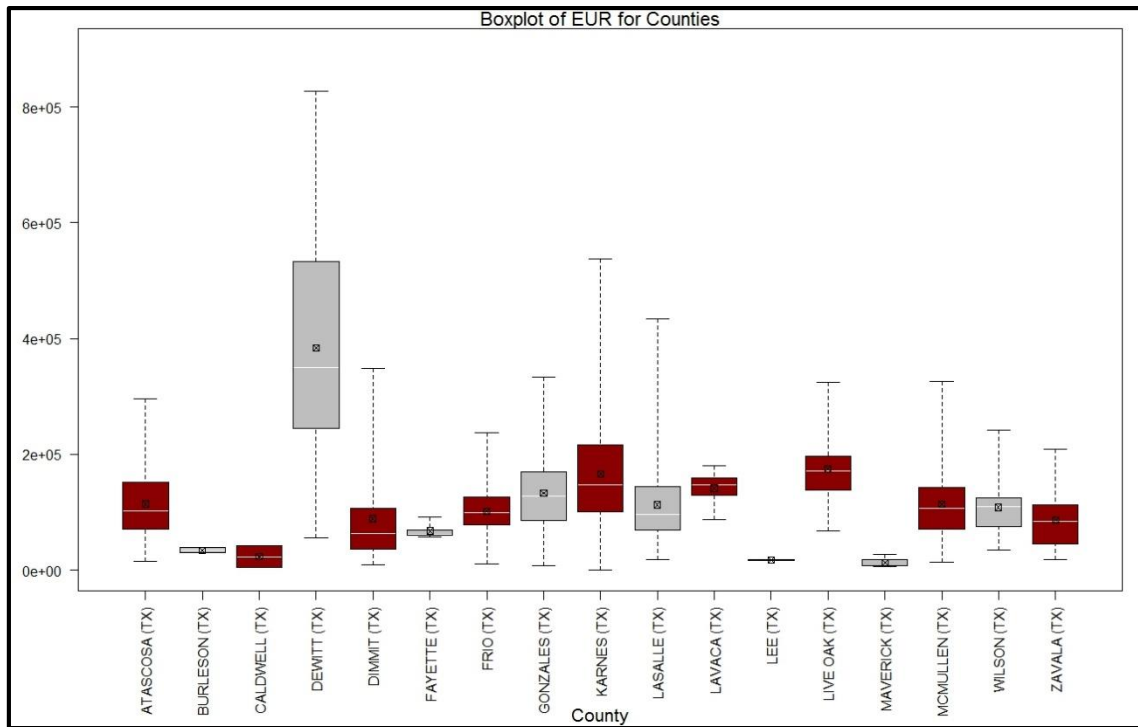


Figure 12. Chart showing box-plots of EUR determined for the oil wells drilled in 2010-11 of the 17 counties of Eagle Ford, Texas.

The boxplots of the calculated EUR are shown (Fig. 12) for all the counties as well as for major operators (Fig. 13) for the wells drilled in the Eagle Ford in 2010-11. The boxplots can be interpreted as shown in Fig. 14.

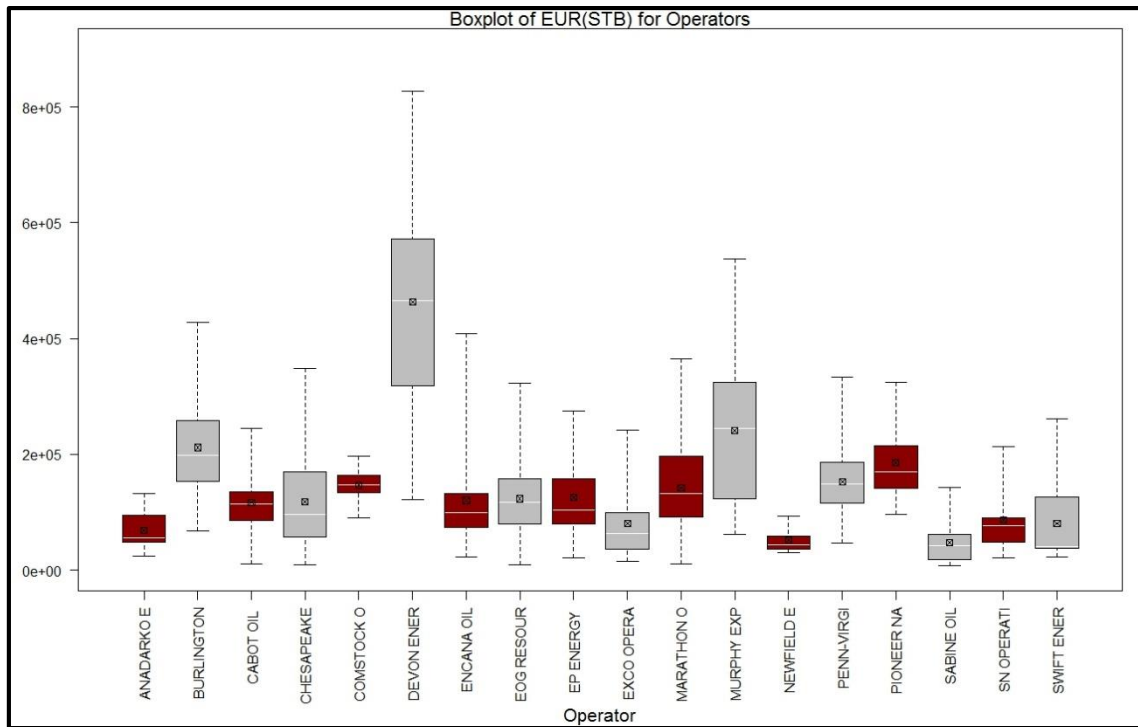


Figure 13. Chart showing box-plots of EUR determined for the oil wells drilled in 2010-11 for the major operators of Eagle Ford, Texas.

The chart showing the boxplots of EUR for different counties (Fig. 12) highlights the distribution of cumulative oil production from different regions of the Eagle Ford. Dewitt County stands out among all to be having one of the best wells. Similarly Devon Energy and Murphy Exploration stand out with the best producing wells in Fig. 13. Due to less data for Burleson, Lee, Caldwell, Fayette and Maverick these boxplots do not appear as per the definition.

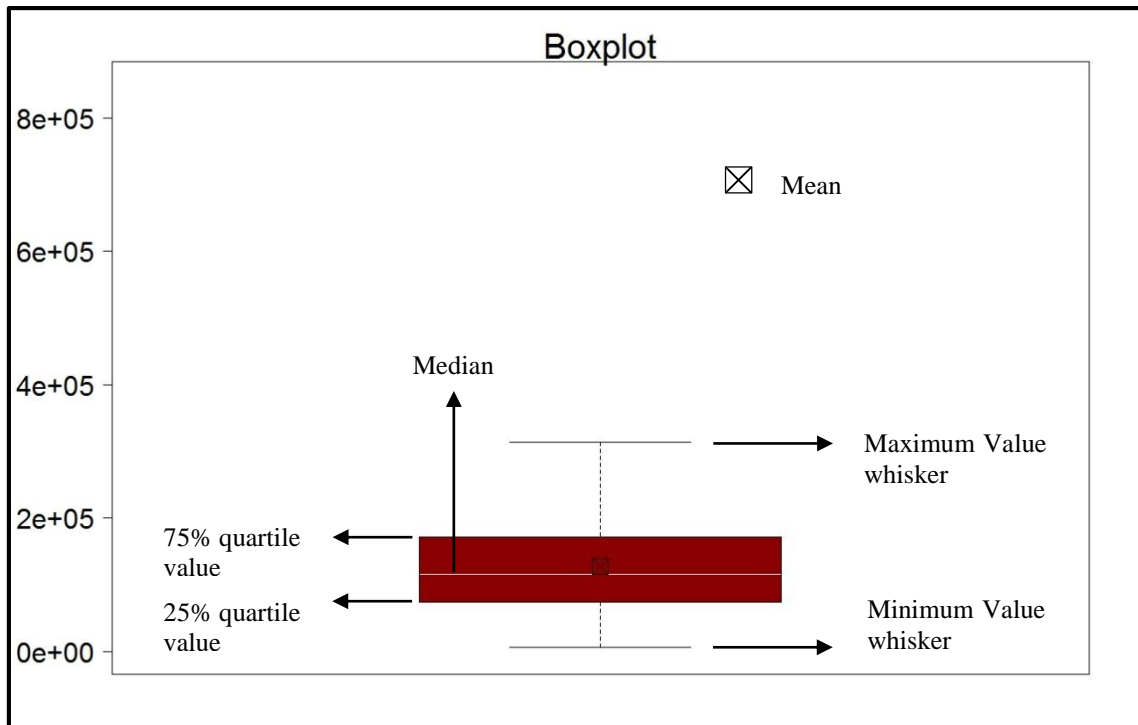


Figure 14. An illustration of how to interpret a boxplot.

CHAPTER IV

EXPLORATORY DATA ANALYSIS

The premier purpose of conducting an Exploratory Data Analysis (EDA) as an intermediate step before jumping on to modeling is to understand the underlying information available with the data. This recognition and interpretation exercise tends to lead us to some useful patterns that exist in the data. These patterns can then be statistically significant or not which can then be proved further by modeling. EDA also helps establish the importance of attributes with respect to the output.

The data consists of 1,134 horizontally drilled, oil producing wells in the Eagle Ford shale. These wells were drilled in 2010-11. The description of the data has been already provided in the third chapter.

In the sections below, the procedure of EDA for this study and its effect on the data has been explained.

4.1 Single Variable Analysis

The advent of digital oil fields and the easier access to data storage and retrieval has taken the oil and gas industry by storm. It is now possible to conduct large amount of data analytics and derive statistical results. Often, interpreting these datasets with so many independent or explanatory variables could be testing. More complexities could arise due to non-normal distribution of these variables which make use of standard statistical testing

difficult. Hence, it is important to take baby steps while interpreting the data. These small conclusions could be then verified or denied by further analysis.

In the single variable analysis, this study tries to establish any interactions existing between the input and the output variables. Table 3 lists the input and output variables that are extracted from DID.

Table 3. List of Input and Output variables considered in this study.

Input/ Explanatory/ Independent Variables	Definitions (DID)	Output/ Dependent Variable
Peak monthly rate (STB/Month)	Peak liquid volume from a well's production's history.	Estimated Ultimate Recovery (EUR) (STB)
Total Depth (MD)	Measured depth that the well was drilled to.	
Liquid Gravity (API)	This is the gravity of the discovery well in the field or when available in the latest report filed with State Comptroller.	
Completed Interval Length (Ft.) (Lateral Length)	The distance between the upper and lower perforations for each well.	
First Month's water production (BBL)	The first month's water production for each well.	
Gas-Oil Ratio (Mscf/ STB)	It is the ratio of last 12 months gas production and last 12 months oil production for a well.	

First of all, any data row, consisting of the input and output variables, which has one or more entry missing is eliminated from the analysis. Most of the statistical functions and models are incapable of handling missing values.

API No	Calculated EUR	County Name	GOR	Liquid Gravity	Operator	Total Depth	First Month Wtr	Lateral Length	Peak Oil
42-013-34278-00	67,146	ATASCOSA (TX)	NA	38.25	ARGENT ENERGY (US) HOLDINGS INC.	13,630	615	3,570	1,725
42-013-34284-00	38,938	ATASCOSA (TX)	468.73	36.1	EOG RESOURCES, INC.	14,220	135	4,512	689
42-013-34318-00	60,767	ATASCOSA (TX)	549.83	40.22	BAYTEX USA DEVELOPMENT, LLC	15,345	946	4,270	6,328
42-013-34321-00	131,558	ATASCOSA (TX)	618.06	44.4	PIONEER NATURAL RES. USA, INC.	16,057	NA	4,832	1,472
42-013-34322-00	117,020	ATASCOSA (TX)	303.14	44	PIONEER NATURAL RES. USA, INC.	15,620	1,987	4,692	1,039
42-013-34323-00	145,896	ATASCOSA (TX)	205.64	42.1	PIONEER NATURAL RES. USA, INC.	14,465	306	3,572	2,020

Missing values due to which the data for API No. “42-013-34278-00” & “42-013-34321-00” needs to be eliminated from analysis.

Figure 15. A screen shot of the CSV data sheet showing a few data rows. The input column of GOR and First month’s water production each show missing values which are then removed from the analysis.

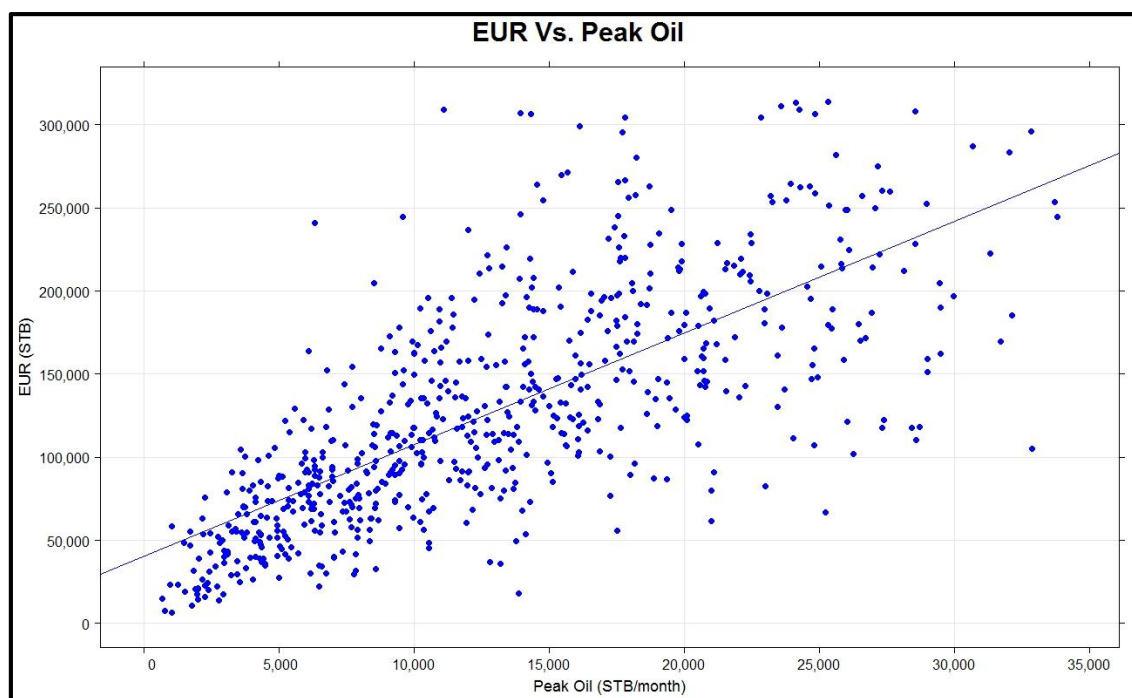


Figure 16. A scatter plot of EUR vs. the peak monthly oil rate for the well with a linear regression line. All the data points are plotted here minus the missing values.

After removing missing values, data points reduce from 1,134 to 989. Fig. 15 shows a depiction of how the missing values affect the removal of a data row. The missing values in GOR as well as first month's water production lead to removal of both the wells from the analysis.

In the first part of EDA, these input variables will be plotted against the output variable and any weak or strong relationship will be examined.

The scatter plot of input variable “Peak oil” and the dependent variable “EUR” shows an acceptable correlation of 0.7254 (Fig. 16). It suggests that increase or decrease in the value of peak oil may have a strong effect on EUR.

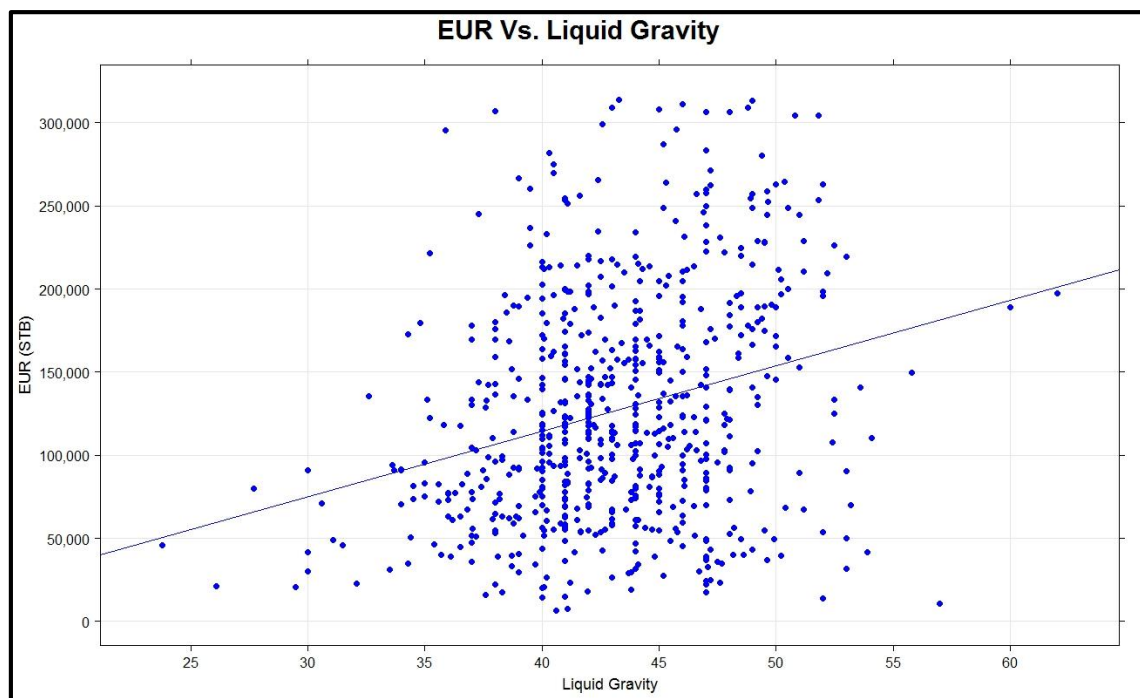


Figure 17. A scatter plot of EUR vs. the liquid gravity for the well with a linear regression line. All the data points are plotted here minus the missing values.

The study conducted by Swindell (2012) concurs with the above result.

The scatter plot of the input variable “Liquid gravity” and the dependent variable “EUR” shows a low correlation of 0.2705 (Fig. 17). Also, careful diagnosis of the plot shows that the linear regression line is highly influenced by extreme data points. It suggests that increase or decrease in the value of liquid gravity may have a weak effect on EUR. The readers are requested to recall the definition of Liquid Gravity provided by DID in Table 3. As the value of liquid gravity may belong to the discovery well of the field, many oil producing wells are prone to possess the same liquid gravity value. Because of this, we see the obvious vertically aligned data points on the plot (Fig. 17).

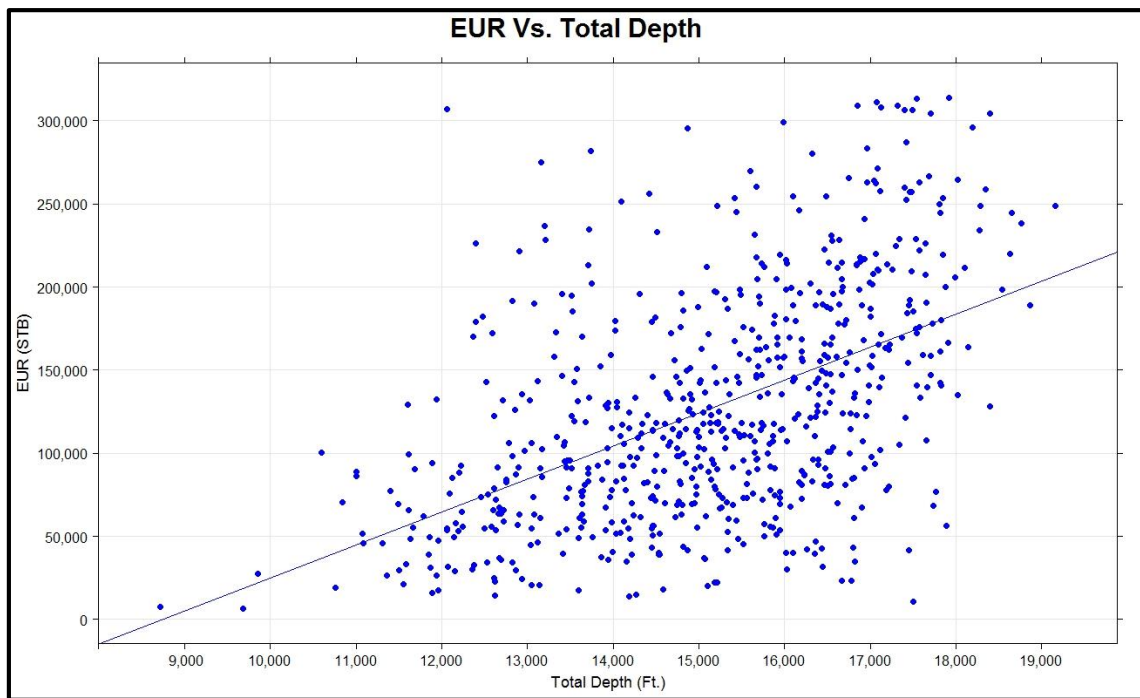


Figure 18. A scatter plot of EUR vs. the Total Depth for the well with a linear regression line. All the data points are plotted here minus the missing values.

The scatter plot of the input variable “Total depth” and the dependent variable “EUR” shows a good correlation of 0.5144 (Fig. 18). It suggests that increase or decrease in the value of Total Depth may have a strong effect on EUR.

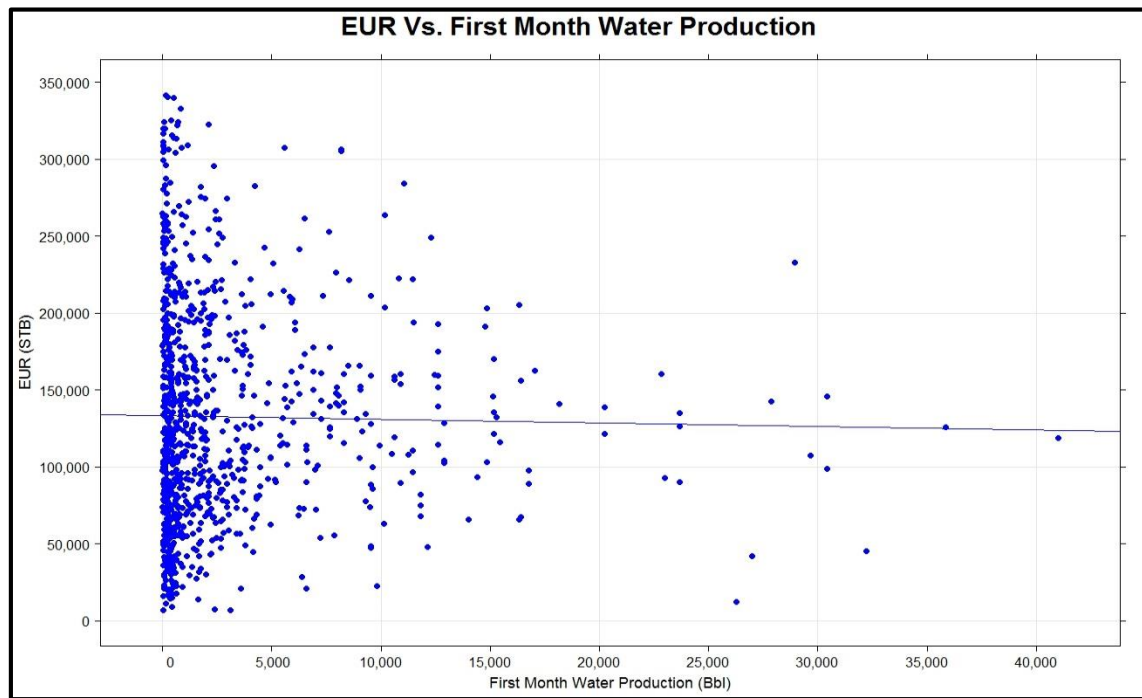


Figure 19. A scatter plot of EUR vs. the First month’s water production for the well with a linear regression line.

The scatter plot of the input variable “First month’s water production” and the dependent variable “EUR” shows a very low correlation of -0.0127 (Fig. 19). It suggests that essentially no correlation exists between the two. The plot may give an impression that most of the value for the input variable are zero. However, it is not so. A closer look of the plot (Fig. 20) shows that.

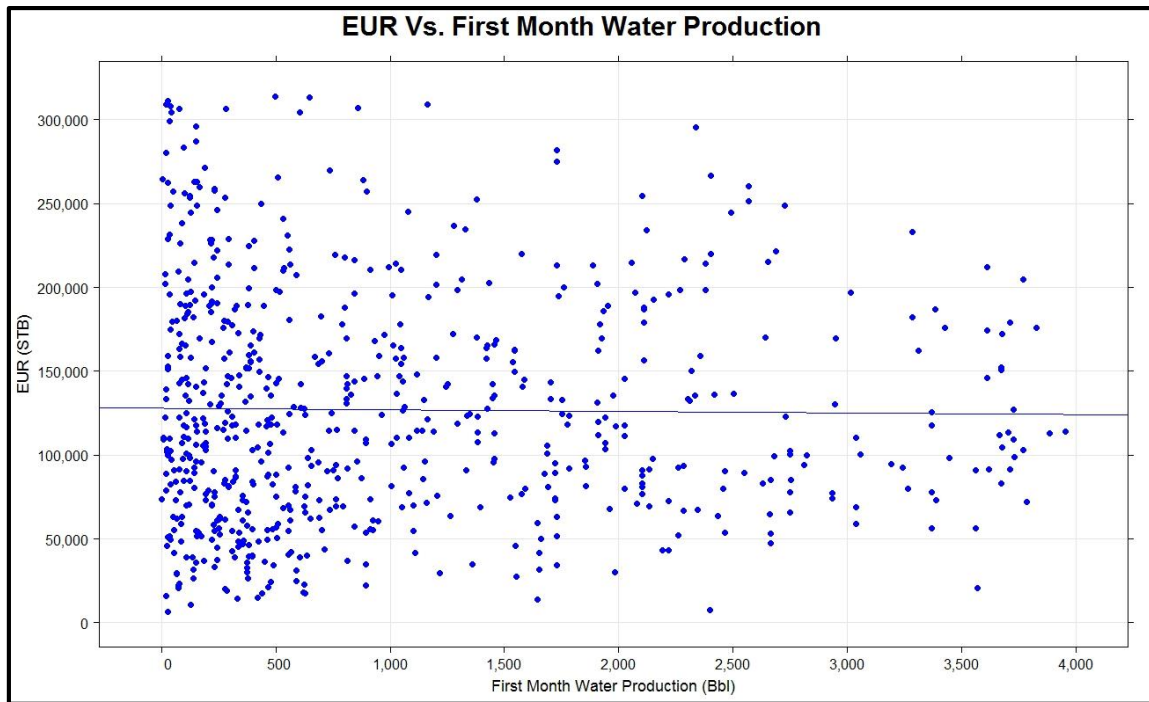


Figure 20. A scatter plot expanding Fig. 19 for the convenience of the reader.

The first look at Fig. 20 may leave an impression of data scattered everywhere and the correlation coefficient also negates any relationship. A deeper study of the data differentiated by County nullifies the above observation. Fig. 21 displays that correlation exists between EUR and First month's water production if data is differentiated by County. The Counties are randomly selected for illustration. The analysis shows that geology of Eagle Ford varies geographically and thus classification of data according to County may improve the modeling.

Encouraged with the above classification results, the data was classified by operators and the scatter plot charted (Fig. 22). Regretfully, no conclusion was drawn. It

would have been helpful if completions data was available from DID. Analysis of which may have distinguished operators from each other.

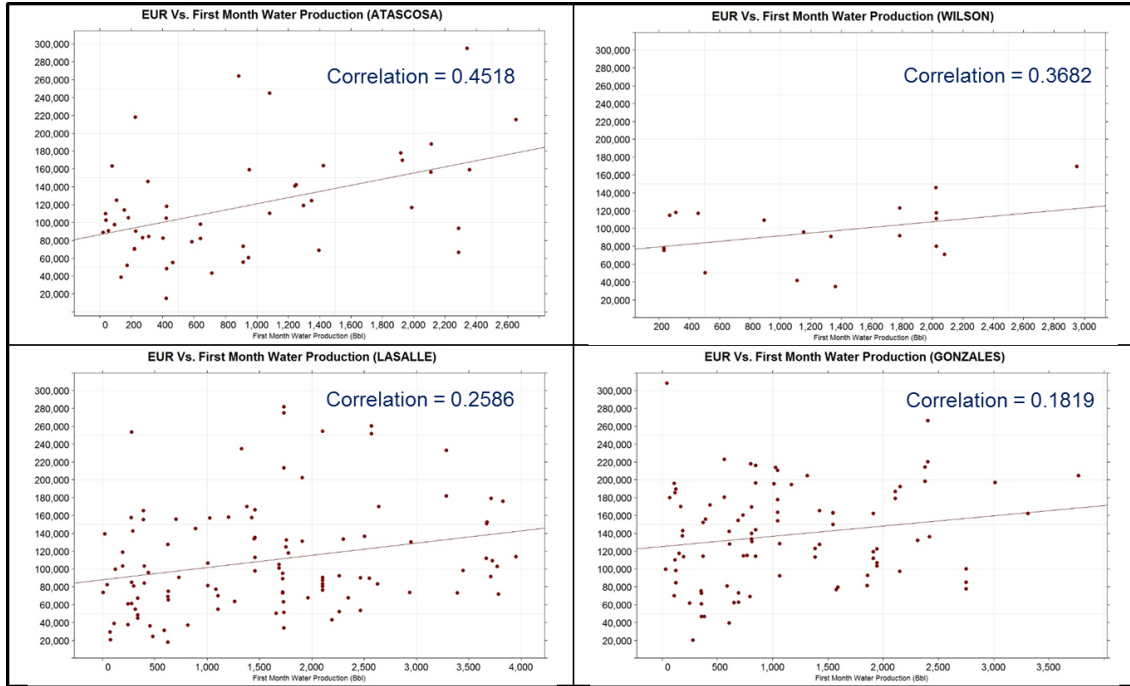


Figure 21. Scatter plots of EUR vs. First month’s water production for Atascosa, Wilson, LaSalle and Gonzales with a linear regression line. The plots also report the correlation coefficient.

The scatter plot of the input variable “Lateral Length” and the dependent variable “EUR” shows a low correlation of 0.1418 (Fig. 23). Thus lateral length may not signify much importance. A closer look at the plot brings out the observation that an optimum value of lateral length exists for obtaining maximum EUR. As Swindell (2012) also suggests, a positive and then a negative correlation exists between the two variables. Thus inclusion of lateral length is necessary in the models.

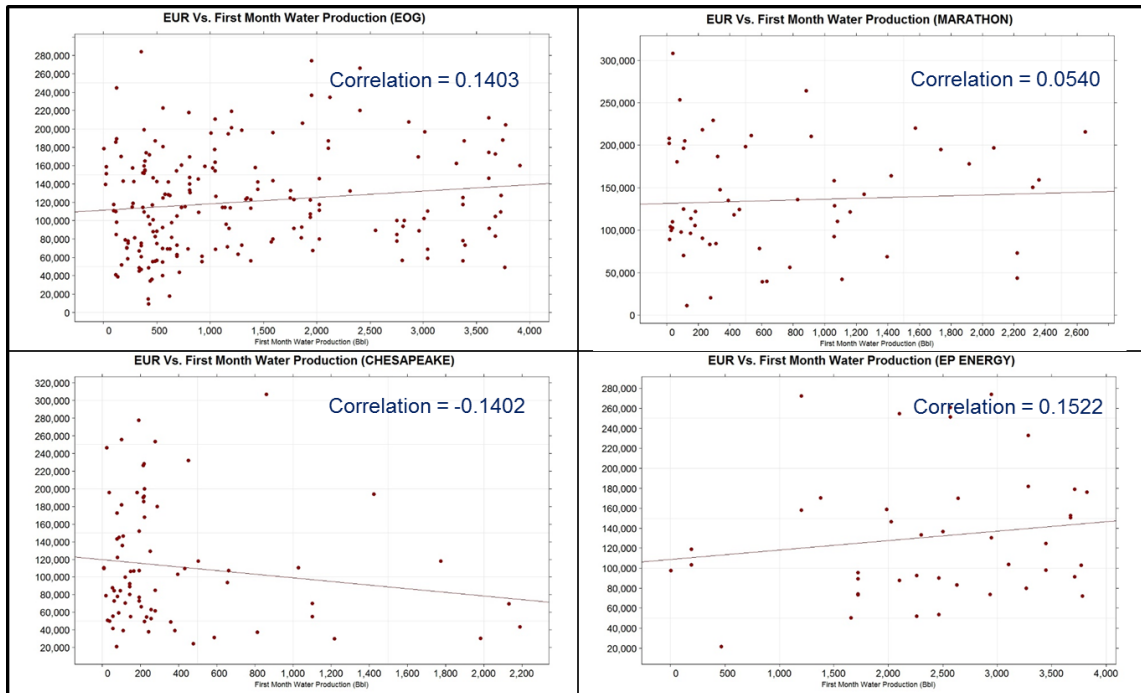


Figure 22. Scatter plots of EUR vs. First month’s water production for major operators with a linear regression line. The plots also report the correlation coefficient.

Similar analysis was performed on lateral length by dividing the data on the basis of County and operators (Fig. 24). It was observed that the correlation coefficients improve on division of the data. All these plots show that an optimum lateral length may exist to maximize EUR. Recklessly drilling with longer completed intervals may not be the correct strategy to maximize recovery.

The scatter plot of the input variable “GOR” and the dependent variable “EUR” shows a low correlation of -0.0267 (Fig. 25 - For clarity, two plots with different GOR ranges are charted.) Thus showing a weak relationship between the input and the output variable.

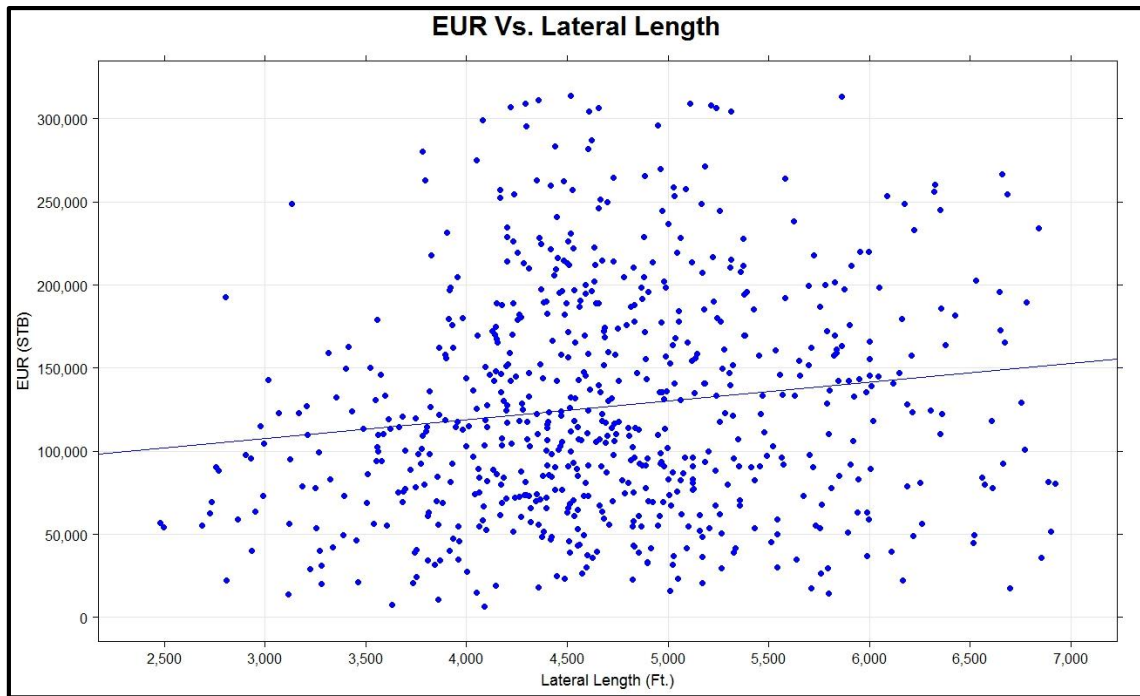


Figure 23. A scatter plot of EUR vs. the completed interval length (Lateral Length) for the wells with a linear regression line.

A complementary analysis was performed on GOR by dividing the data on the basis of County and operators (Fig. 26). It was observed that unlike “First month’s water production” and “Lateral length”, there was not much improvement in the correlation coefficients of “GOR” on division of the data. It goes along with the understanding that this study tries to predict oil’s EUR which doesn’t include any monthly production of gas and hence GOR should not impact EUR considerably.

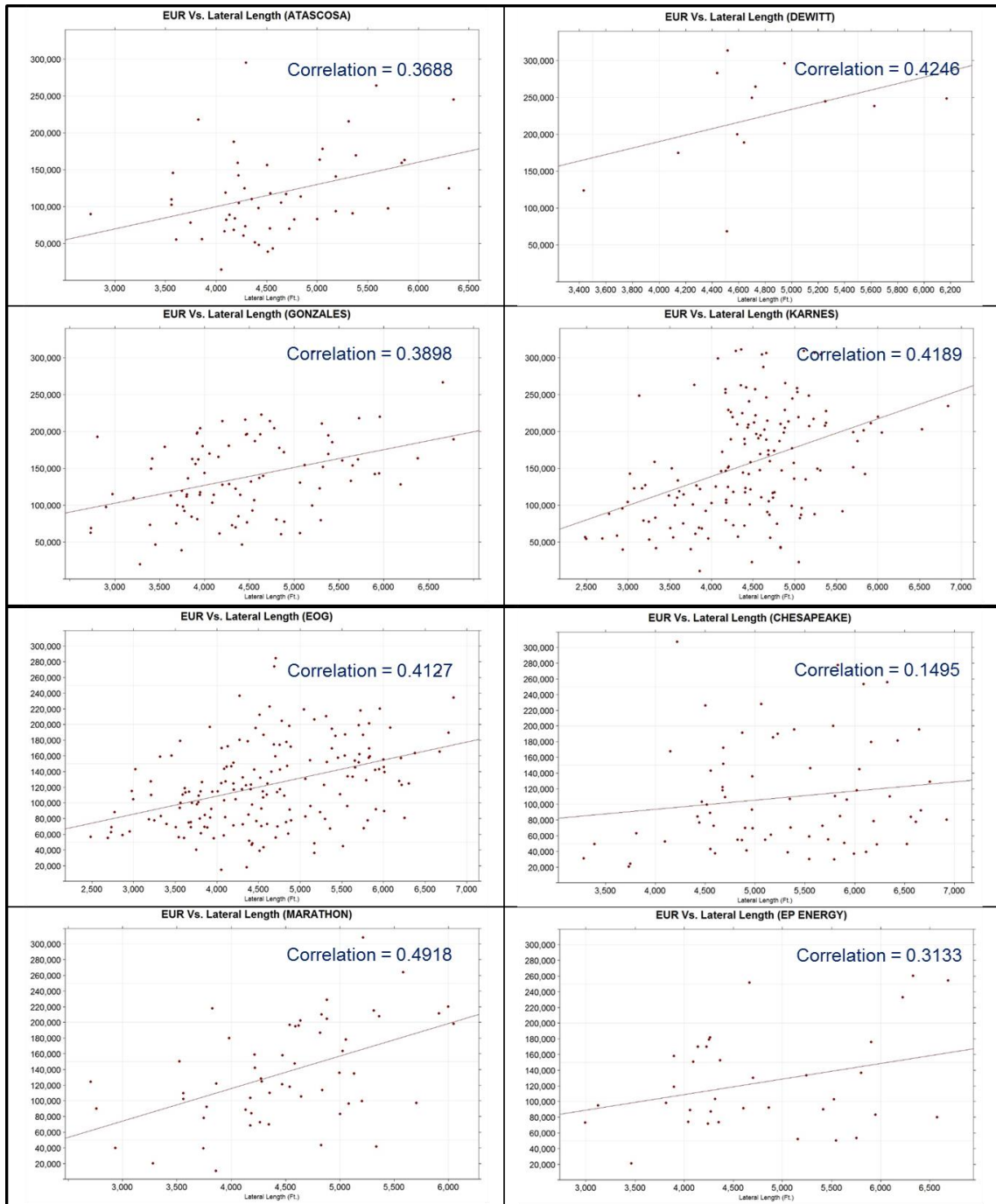


Figure 24. Scatter plots of EUR vs. Lateral length for Atascosa, Dewitt, Gonzales, Karnes County and major operators with a linear regression line. The plots also report the correlation coefficient.

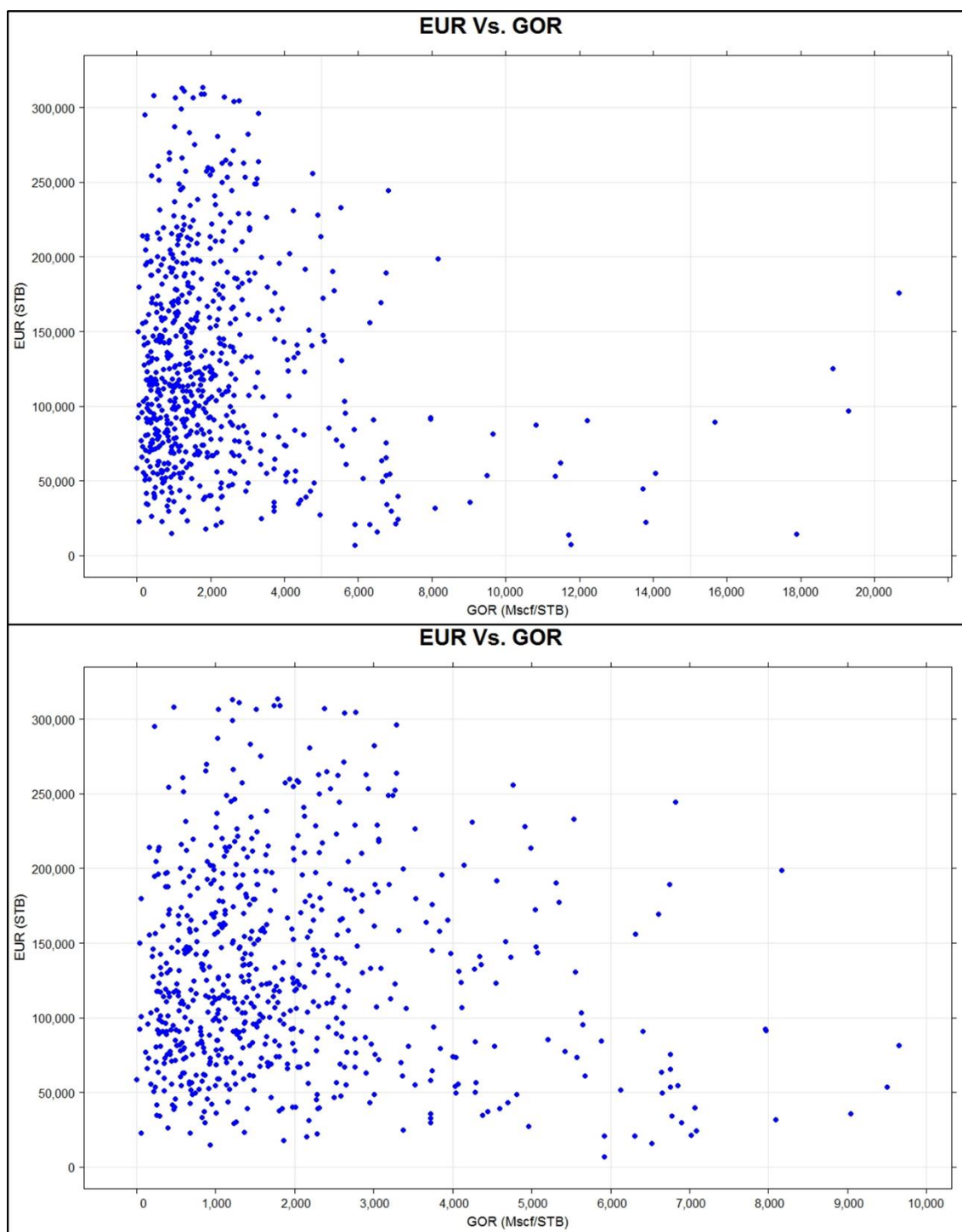


Figure 25. Scatter plots of EUR vs. Gas-Oil ratio for the wells. The top Plot shows GOR range till 20,000 Mscf/STB whereas below one displays a deeper look.

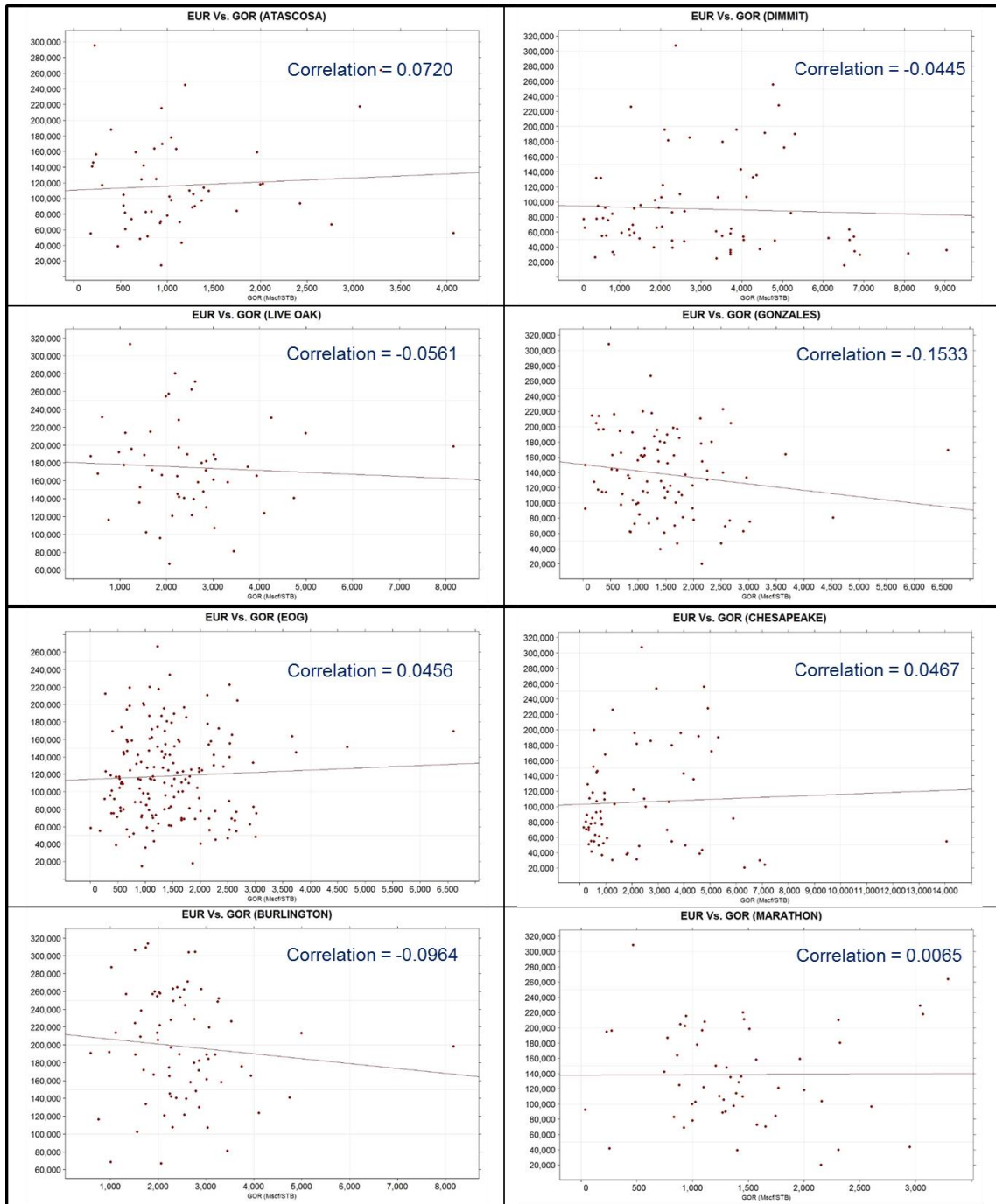


Figure 26. Scatter plots of EUR vs. Gas-Oil ratio for Atascosa, Dimmit, Live Oak, Gonzales County and major operators with a linear regression line. The plots also report the correlation coefficient.

On the basis of the above discussion about the relationships between individual input and output variables, 5 inputs were selected for further analysis. The Table 4 lists these variables. The input variables are listed in decreasing order of their importance.

Table 4. List of final input and output variables for analysis.

Input/ Explanatory/ Independent Variables	Output/ Dependent Variable
Peak monthly rate (STB/Month)	Estimated Ultimate Recovery (EUR) (STB)
Total Depth (MD)	
Liquid Gravity (API)	
Completed Interval Length (Ft.) (Lateral Length)	
First Month's water production (BBL)	

4.2 Outlier Analysis

Extreme or dominant data point are termed as outliers. There has been no clear procedure laid out to analyze and remove these points from the study, hence supreme caution is advised for outlier analysis and subsequent removal. These extreme data points may interfere with the analyses of the data significantly. The existence of outliers may lead to exaggerated errors and consequential wrong estimates of statistical results.

Hawkins described an outlier as an observation that “deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism” (Hawkins (1980)). These points lie near three standard deviations from the mean and hence may have a serious effect on statistical analyses. Anscombe and Guttman (1960) identify outliers as arising from errors in data reporting or those arising from the original

distribution of the data. Outliers tend to bias estimates, increase error variance and decrease normality of the data.

In this study, outliers are detected and removed using various statistical tests which are listed as below:

1. By following a simple thumb rule of $\bar{x} \pm 3 s.d.$
2. By checking the normality of the data using Quantile-Quantile (Q-Q) plots.
3. Box-plots.

Firstly, the outliers were detected using the simple rule of $\bar{x} \pm 3 s.d.$ and the normality of the data was checked with the Q-Q plots. With the removal of each data point identified as an outlier, the normality of the data should be restored. The boxplots come in very useful while plotting the outliers along with the whole distribution. Boxplots display outliers as points outside the distribution.

The said investigation is done on all the input and output variables and their boxplots are plotted in Fig. 27 & Fig. 28. Of all the 6 parameters, few or all outliers are removed from all but “Total Depth” and “Liquid Gravity”. Both these parameters do not show any improvement in correlation with EUR, normality or fit of the model on these points removal. It is to be understood that when an outlier is removed from any variable, the whole corresponding data row is deleted just like it happens when missing data is removed.

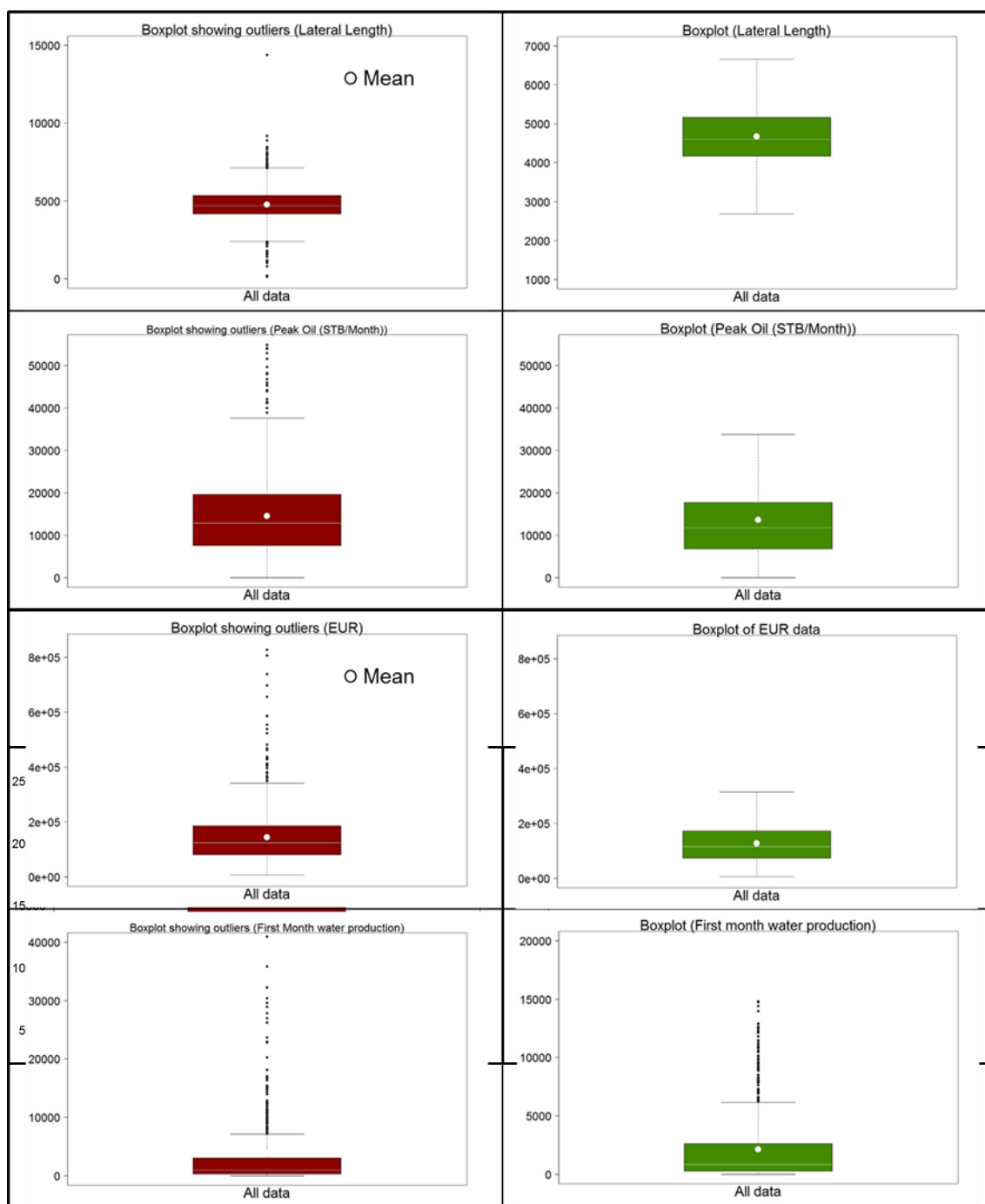


Figure 27. Boxplots of various input and output variables before and after outlier analysis.

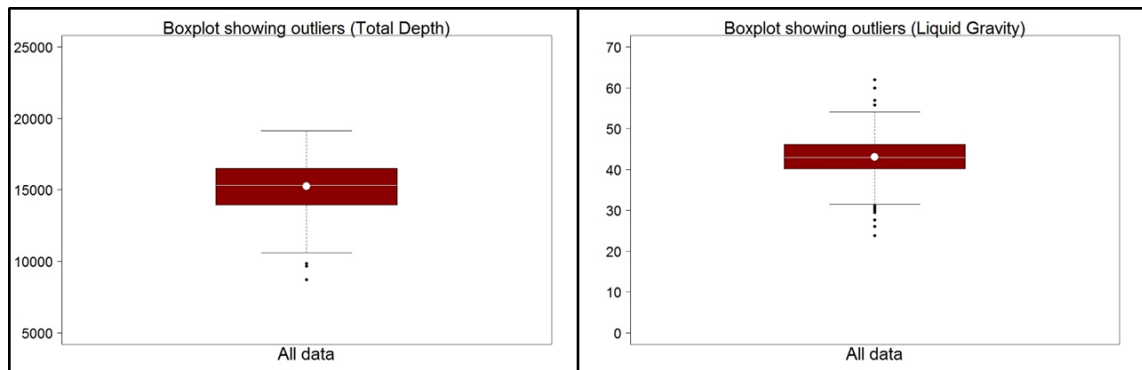


Figure 28. Boxplots showing outliers for “Total Depth” and “Liquid Gravity” input variables.

In the process, outliers were removed and the data set was reduced to 833 points belonging to 12 counties. Further modeling and prediction is performed on this cleaned dataset.

CHAPTER V

MODELING AND RESULTS

After establishing qualitative relationships between input and output variables and cleaning the data of deleterious outliers, MLR, SVM and BRNN modeling is done and results compared.

5.1 Methodology and Results from Modeling

The exploratory data analysis conducted in Chapter IV helped to define the relationships between the variables. It was also seen that grouping the data into different categories was particularly important in establishing any meaning relationships in few variables. This grouping of data is also called clustering.

Clustering could be performed using various algorithms in such a way that data point in one group is more similar to other data points in the same group rather than in the other group.

Taking cues from the clustering process, three data sets were prepared for modeling purpose as listed in Table 5. The algorithmic clustering is done using *k-means* algorithm from the *stats* package in R. There are many algorithms which could be used to perform this task but the one used in R is proposed by Hartigan and Wong (1979). It usually does a better job than other algorithms (R Core Team (2015)). The data is partitioned into k groups such that the sum of squares from points to the assigned cluster

centers is minimized. At the minimum, all cluster centers are at their corresponding group's mean (R Core Team (2015)).

Table 5. List of data sets used in modeling.

Data set	Explanation
Whole Data	All the 833 data points that the clean data contains belong to this data set.
County Clusters	All the 833 data points are divided into groups of Counties.
Algorithmic Clusters	All the 833 data points are divided into 6 clusters using k-means algorithm.

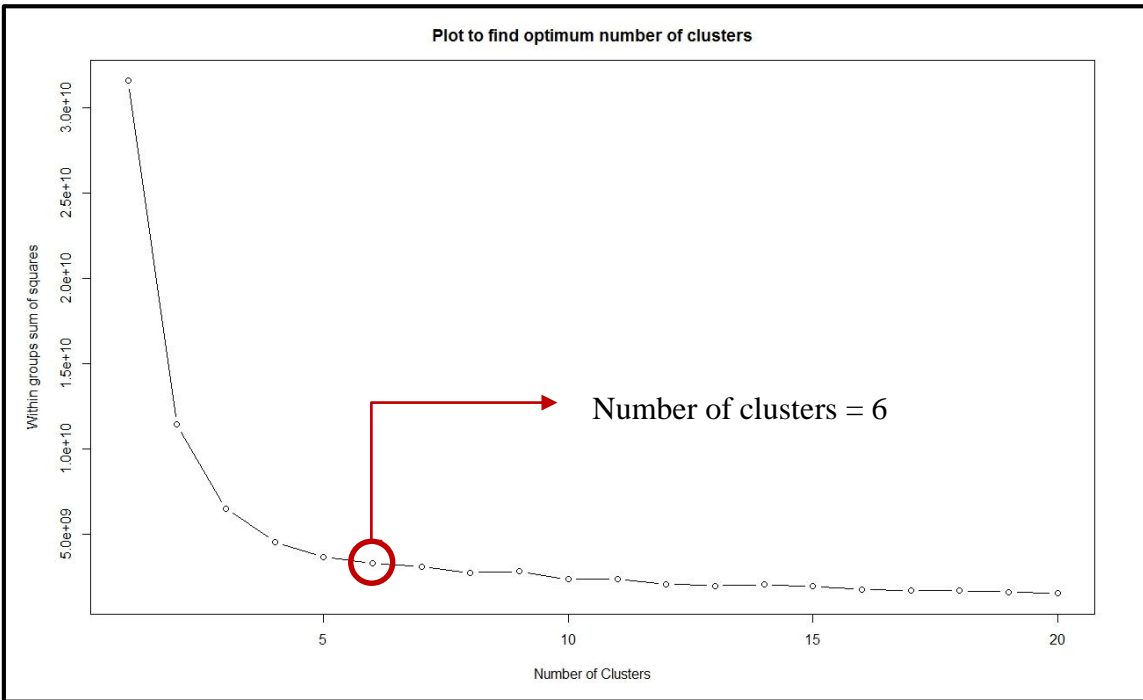


Figure 29. A plot showing how the optimum number of clusters are determined.

The optimum number of clusters is obtained by plotting the total sum of squares and the number of clusters (Fig. 29). The number of clusters is so chosen that total sum of squares is minimum and further increase in cluster numbers do not impact the sum of squares much. With careful observation, six number of clusters seem optimum in this case. It is to be noted that all the input variables are used to form these clusters.

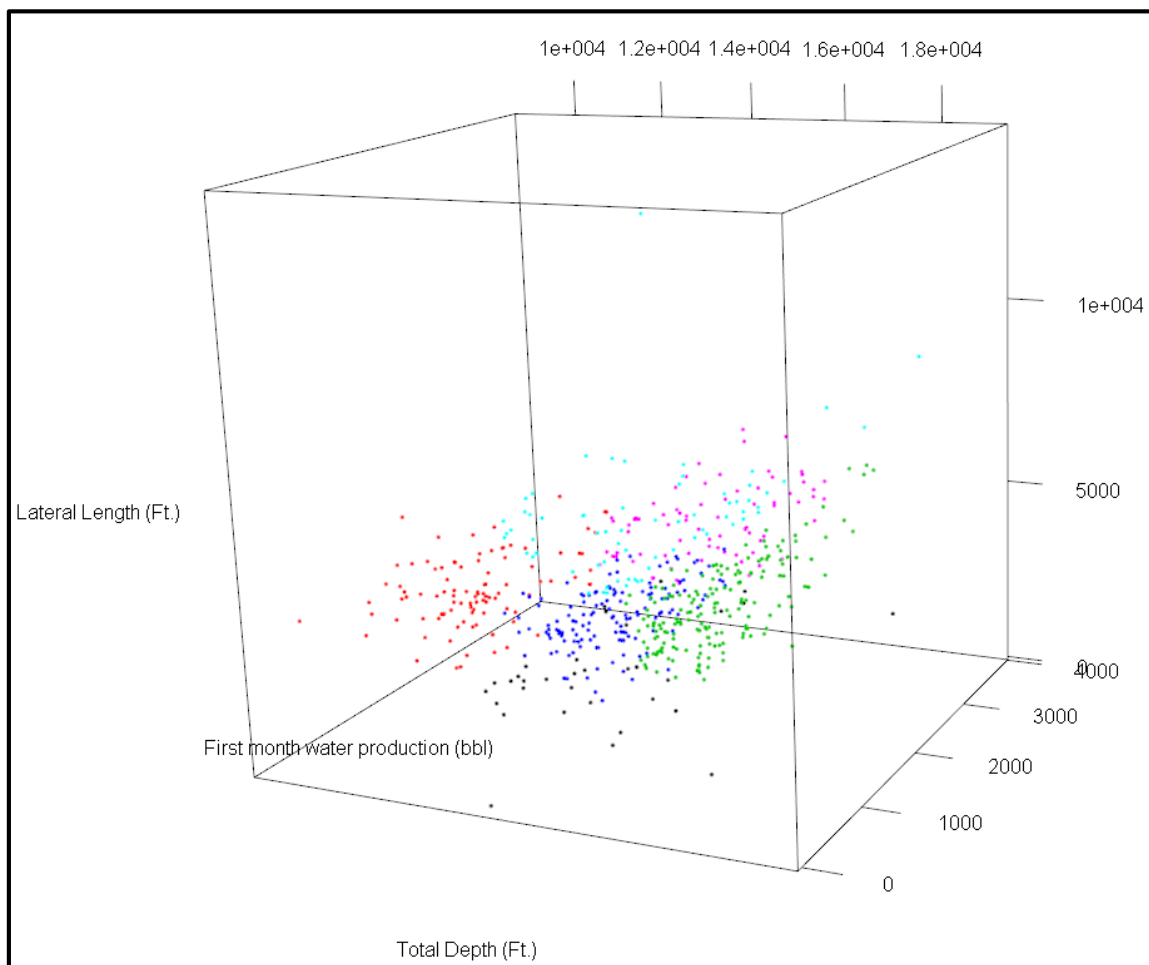


Figure 30. A 3D plot showing 3 input variables clustered using k-means algorithm. The clustering is done on all the 5 inputs.

For visualization and easier understanding of the reader, a sample 3D plot (Fig. 30) is produced showing “Total Depth”, “First month’s water production” and “Lateral Length” clustered into six groups. It is to be noted that although the clustering is done using all the 5 inputs, it is visually easy to understand a 3D plot. A total of 10 plots like Fig. 30 would be necessary to visualize all the 5 inputs.

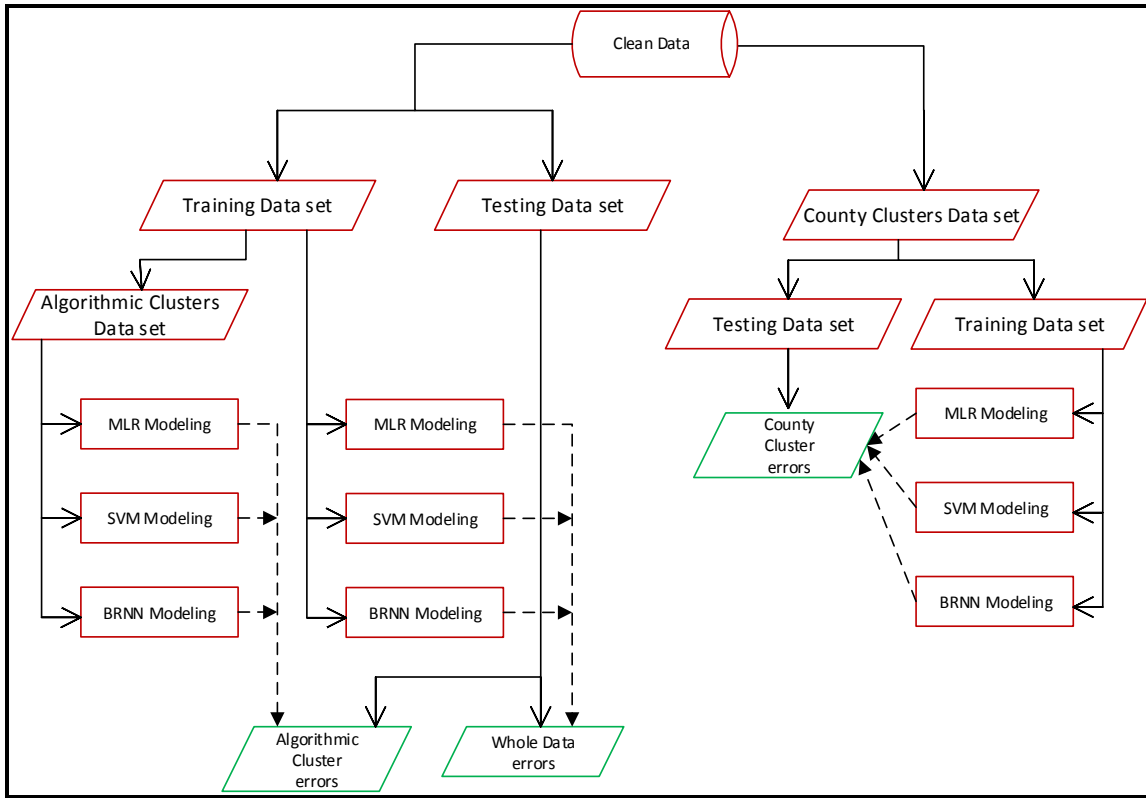


Figure 31. A flowchart depicting the progress of the modeling process.

The modeling process has been explained through the flowchart in Fig. 31. The critical path is elucidated as below:

1. Cleaned data is obtained after exploratory data analysis.

2. This cleaned data is then divided into groups of similar counties known as County clusters.
3. Each County cluster is then divided into random train and test data. Training and testing data set is divided into 80:20 ratio.
4. All the MLR, SVM and BRNN models are created using the train data set and predictions are made on the testing data. All the errors are stored as County cluster errors.
5. Furthermore, cleaned data is divided into training and testing data set with 80:20 ratio randomly.
6. Two calculations are performed on the training data set. First, MLR, SVM and BRNN models are created using this data. Whole data errors are evaluated by predicting on these models using the testing data set.
7. Second, this training data set is divided into six algorithmic clusters using *k-means* function in R. MLR, SVM and BRNN models are produced on each of these cluster.
8. Each data point in the testing data set is identified to be in one of the six clusters. Predictions are then performed on the testing data set and algorithmic cluster errors are calculated.

BRNN modeling is sensitive to scale of the inputs (Perez-Rodriguez et al. (2013)), hence all the inputs are scaled to $[-1,1]$. In all the three data sets, normalization is done on the input and output variables with respect to the training data set and the normalization parameters retained. While predicting on the inputs of testing data, the variables are

normalized using the training data normalization parameters. After prediction, the output is rescaled to the original. Also, the input and output variables of the training data set are rescaled back to original.

For the comparison of errors, the median percentage absolute error metric is utilized. The standard statistical measure for the median percentage absolute error is defined as:

$$MAE = Median \left(100\% \left| \frac{A_i - P_i}{P_i} \right|, i \rightarrow [1, n] \right)$$

Where, n is the length of test data, A_i & P_i are the actual and predicted output variable.

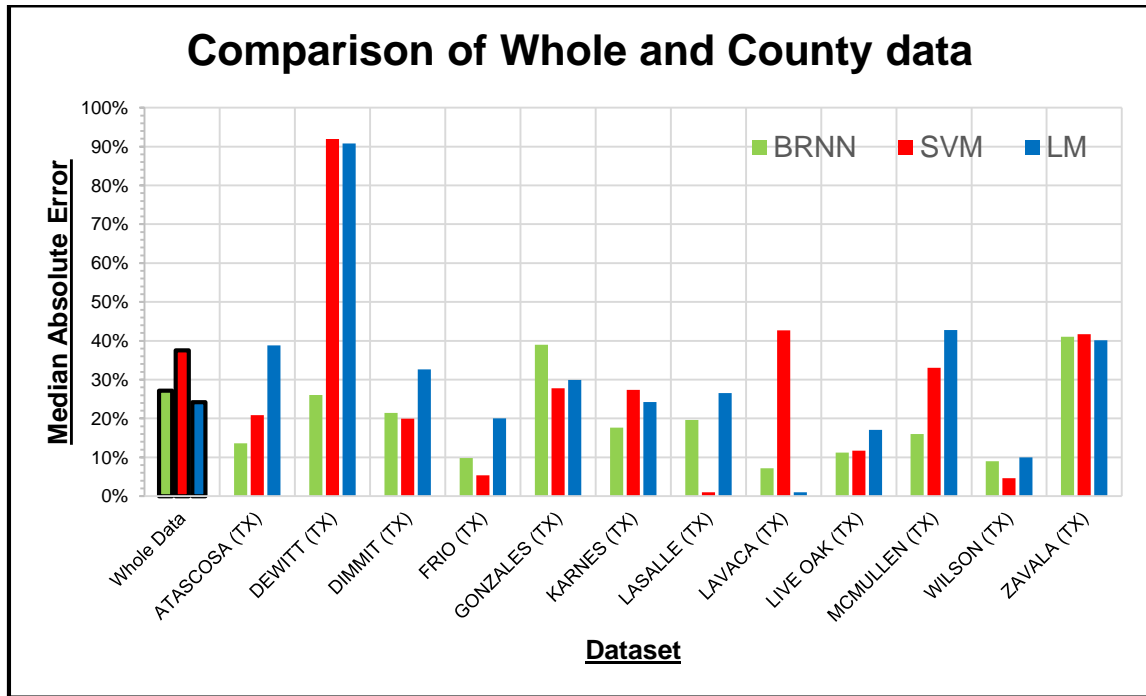


Figure 32. Comparison of median percentage absolute errors for the whole data and the County clusters. LM stands for Linear Model (MLR).

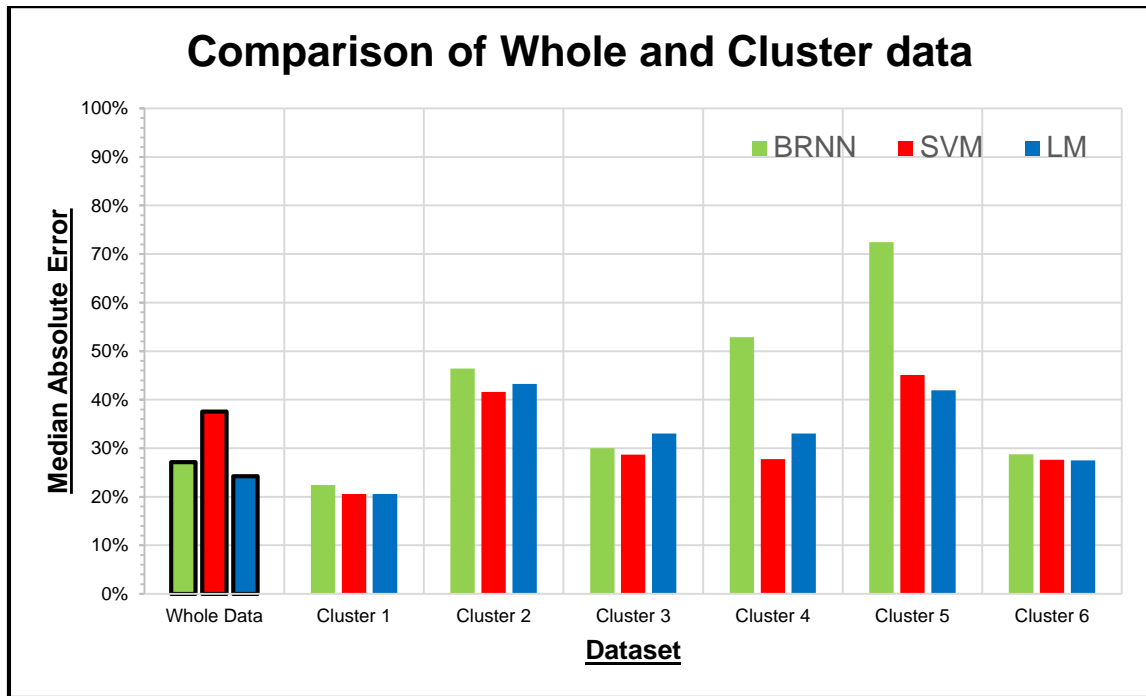


Figure 33. Comparison of median percentage absolute errors for the whole data and the algorithmic clusters. LM stands for Linear Model (MLR).

The median percentage absolute errors for all the data sets and the models are reported in the Fig. 32 and Fig. 33. The errors for whole data are utilized as an anchor for connecting County and algorithmic clusters. The errors have variations due to the approach used for modeling, the number of data points in the training set, the distribution of input variables and the inherent errors in the data itself. Attaching one reason or the other to any model to explain its behavior is difficult. Especially for neural networks which are rather a black box.

5.2 Model Selection Algorithm

Having 19 (1 whole + 6 Algorithmic clusters + 12 County clusters) different data set models and each data set model containing 3 median absolute percentage errors could be overwhelming for the user. For this purpose a model selection algorithm was coded in R. The end result the algorithm seeks is one error which defines the comparison and usability of these three modeling methods.

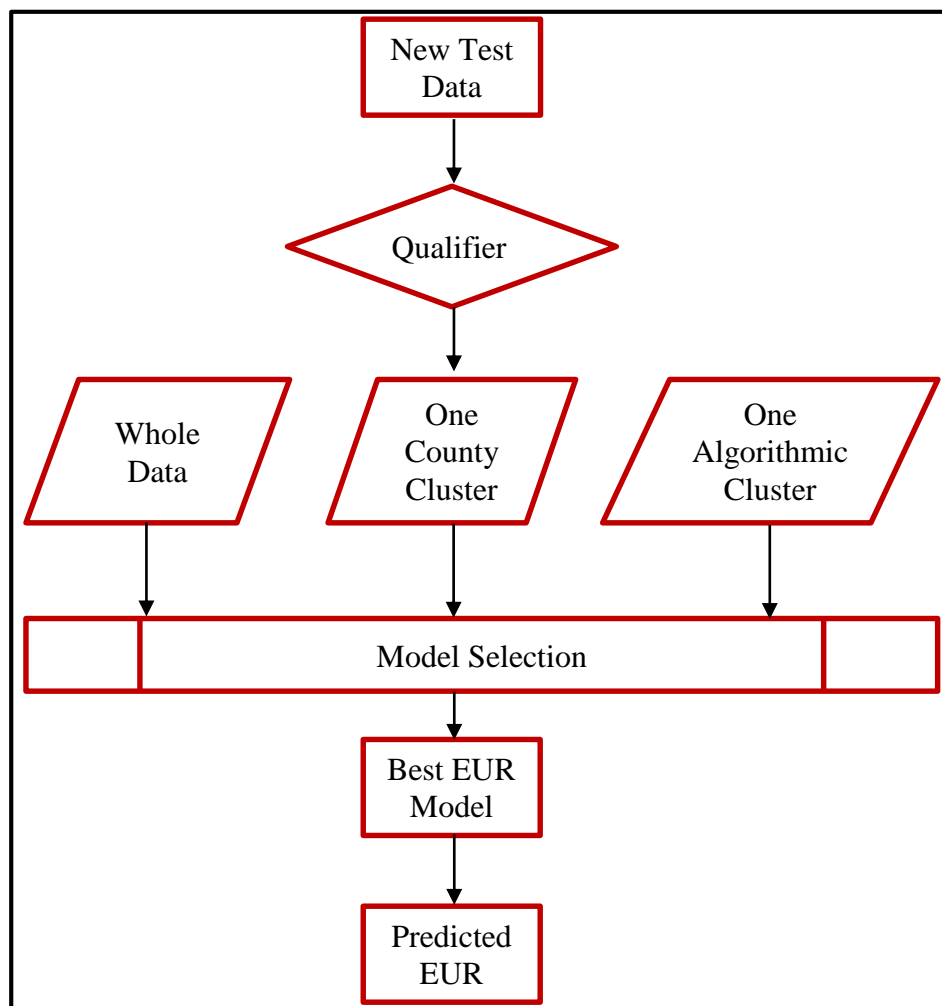


Figure 34. Flow chart depicting the process of model selection.

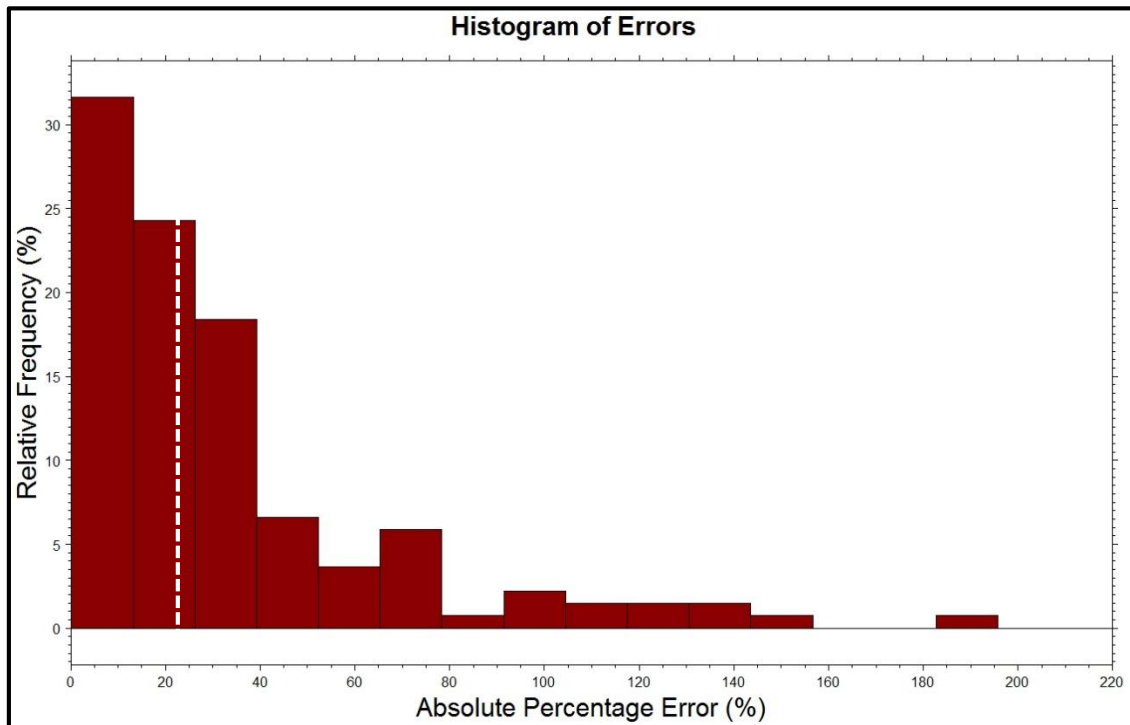


Figure 35. A histogram of Absolute percentage errors for the testing data obtained from the model selection algorithm. The median is shown with the use of white dashed line at 22.00%.

A schematic is drawn (Fig. 34) to elucidate the process followed in the model selection algorithm. Each new data point belongs to the whole data set, one County cluster and one algorithmic cluster. The qualifier does the identification of such new data point. Best models had already been selected on the basis of median percentage absolute errors for the 19 data sets (Fig. 32 & Fig. 33). To explain the process, let's take an example:

Let's say a new data point belongs to the whole data, Karnes County and Algorithmic cluster 4. All these three data sets have MLR, SVM and BRNN models' median percentage absolute errors. The best of these models are selected. In this particular

case, MLR model for whole data, BRNN model for Karnes County and SVM model for cluster 4 are selected. Comparing the errors for these three models, the BRNN model of Karnes County is selected to be the best model in this case. Finally, EUR prediction is achieved using this model. Furthermore, all the testing data set is sent through the model selection algorithm and errors are calculated.

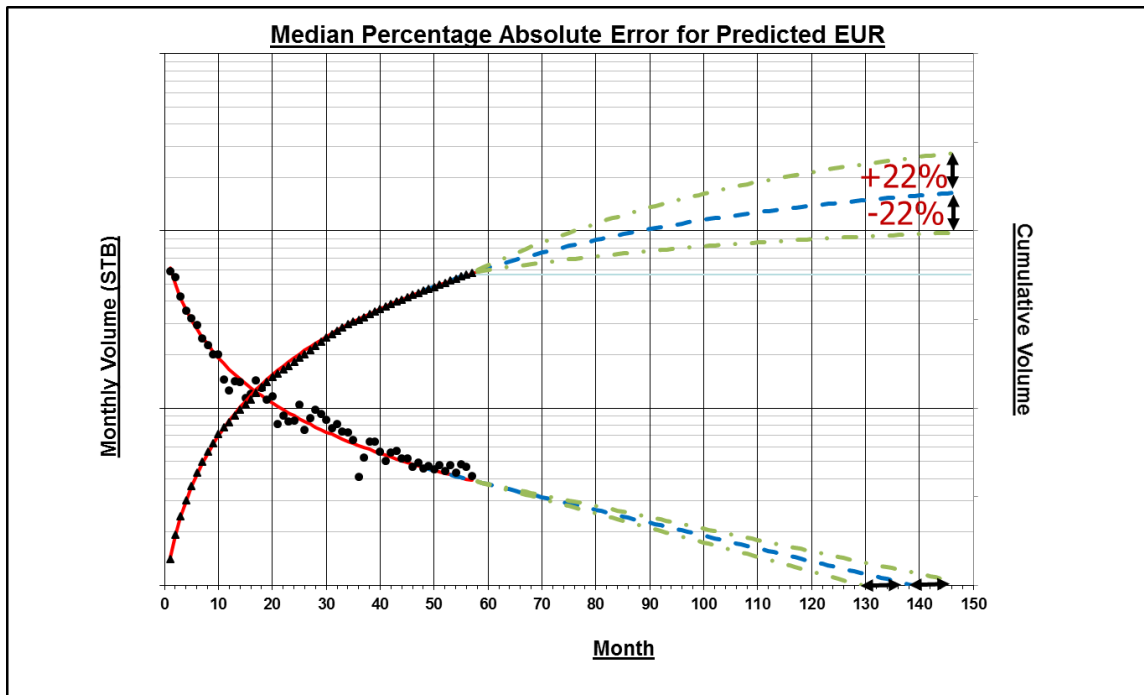


Figure 36. An illustrative plot of decline curve showing the range of median percentage error in the prediction of EUR using the model selection algorithm.

The median percentage absolute error for the test data sent through the model selection algorithm is 22.00%.

CHAPTER VI

DISCUSSION AND CONCLUSIONS

The objective of the study was to explore and apply suitable machine learning algorithms for the prediction of EUR. Conventionally, production data is graphically extrapolated to an abandonment rate resulting in expected cumulative production. Using machine learning predictions is useful as it won't require production data and hence would be a useful decision making tool especially in today's low price environment.

In this study, actual EUR is determined using Arps' decline curve analysis (economic limit: 150 STB/month) and then compared with predicted EUR determined using machine learning methods. The closer they are, the better our prediction models.

The production data collected for this study is from 2010-11 horizontal wells of the Eagle Ford producing primarily oil. All the production and well data is extracted from Drillinginfo desktop application. There are severe limitations to what information is available through public data and thus the modeling is done using five input variables:

1. Peak monthly oil volume
2. Total depth
3. Liquid gravity
4. Completed interval length (lateral length)
5. First month's water production

It is important to pre-process the data in order to obtain optimum training and testing data set. The pre-processing stage is called exploratory data analysis. Evaluating

relationships between input and output variables, discarding any ineffective parameter and outlier analysis are part of this pre-processing. In this study, gas-oil ratio was removed from the list of inputs following exploratory data analysis. Furthermore, after removing missing values and outliers the data points are reduced to 833 from 1,134. Of these, 145 were missing values.

Any inconspicuous heterogeneity in the data is taken into account by sub-division of data geographically and algorithmically. Thus, the data is divided into clusters; County wise and using *k-means* algorithm. The optimum number of algorithmic clusters was evaluated to be six. Also, there were 12 County clusters and 1 whole data set. In all, there were 19 data sets on which modeling using Multiple Linear Regression, Support Vector Machine (SVM) and Bayesian Regularized Neural Networks (BRNN) was done.

For BRNN, it is important to scale the training and testing data set to $[-1, 1]$. The normalization parameters used are from the training data set. It helps stabilizing the sensitivity of neural networks.

Modeling on all the 19 datasets with 3 methods yielded 57 models. Combining errors from these many models was a challenge and a model selection algorithm was formulated to solve it. The median percentage absolute error of 22% shows that machine learning algorithms could be of immense usefulness for predicting EUR.

These are the primary conclusions that can be drawn from this study:

1. Plotting boxplots of EUR distinctly indicate which counties and operators are performing better (Fig. 12 & 13).

2. The strongest and weakest correlation of EUR was found to be with peak oil and GOR respectively (Fig. 16 & 25).
3. A weak correlation was observed between EUR and completed interval length because of presence of optimum interval of 4,000 to 6,000 ft. (Fig. 23).
4. Correlations may improve between EUR and input variables after subdividing the data by county (Fig. 21, 22, 24 & 26).
5. Most of the best performing models belong to machine learning methods (Fig. 32 & 33). Due to non-linearity of data, multiple linear regression is inadequate for robust prediction of EUR.
6. Combining the models' errors was achieved through model selection algorithm which yielded a median percentage absolute error of 22%. Thus half of the errors lie below 22%.

The whole programming is coded in R language. Thus data extraction, data preparation, exploratory data analysis, cluster analysis, modeling, charting of the data, model selection algorithm are reproducible with minor changes. Provided with sufficient amount of production data the process can be repeated for wells drilled in any year and field. In fact, the prepared models could be used as well but we'll be assuming that the production and completion technology is at the same level as of 2010-11.

Similar study could be done for the gas wells of any field if production data is made available in optimum amount.

There are several other supervised and unsupervised machine learning methods which could be tested and utilized for enhancing the study. Addition of some more methods may bring the error considerably down.

EUR determination is done using Arps' decline curve analysis which requires the well to be producing in boundary dominated flow. Other approaches like Power law exponential method, Stretched exponential production decline method, Duong's method or Logistic growth model method could be used for EUR determination for shale wells.

A piece of code is also written in R which will help investigators to implement the current models to predict on a new data point. The investigator would need all the 5 inputs and the County which will then predict EUR using whole data set model, County model and algorithmic cluster model. The code will automatically place the new data in one of the six clusters. The result would be predicted EUR from all the nine models.

This study is a further step in the direction of data-driven problem solving approach in oil and gas industry. It could very well be used as a decision making tool in the planning stage of a well.

REFERENCES

- Anscombe, F.J. and Guttman, I. 1960. Rejection of Outliers. *Technometrics* **2** (2): 123-147. DOI: 10.2307/1266540
- Arps, J.J. 1945. Analysis of Decline Curves Original Edition. ISBN 0096-4778
- Bhatt, A. 2002. Reservoir Properties from Well Logs Using Neural Networks. *Oslo: Department of Petroleum Engineering and Applied Geophysics Norwegian University of Science and Technology*.
- Chen, B., Kumar, D., Uerling, A. et al. 2015. Integrated Petrophysical and Geophysical Analysis on Identifying Eagle Ford Sweet Spots. Society of Petroleum Engineers. DOI: 10.2118/178609-MS.
- Cortes, C. and Vapnik, V. 1995. Support-Vector Networks. *Machine Learning* **20** (3): 273-297. DOI: 10.1007/bf00994018
- Crnkovic-Friis, L. and Erlandson, M. 2015. Geology Driven Eur Prediction Using Deep Learning. Society of Petroleum Engineers. DOI: 10.2118/174799-MS.
- Gao, C. and Gao, H. 2013. Evaluating Early-Time Eagle Ford Well Performance Using Multivariate Adaptive Regression Splines (Mars). Society of Petroleum Engineers. DOI: 10.2118/166462-MS.
- Hartigan, J.A. and Wong, M.A. 1979. A K-Means Clustering Algorithm. *Applied Statistics* 28: 100-108.
- Hawkins, D.M. 1980. *Identification of Outliers*. Monographs on Applied Probability and Statistics; Monographs on Applied Probability and Statistics. London ;; Chapman and Hall. Original edition. ISBN 041221900X 9780412219009.
- Hsu, C.-W., Chang, C.-C., and Lin, C.-J. 2010. A Practical Guide to Support Vector Classification, Department of Computer Science, Ntu, Taiwan.

- Jones, V. 2015. Swift: More Than 150,000 Jobs Lost in Oil, Gas. http://www.rigzone.com/news/oil_gas/a/139230/Swift_More_than_150000_Jobs_Lost_in_Oil_Gas
- Masters, T. 1993. Practical Neural Network Recipes in C++. ISBN 9780124790407
- Perez-Rodriguez, P., Gianola, D., Weigel, K.A. et al. 2013. An R Package for Fitting Bayesian Regularized Neural Networks with Applications in Animal Breeding. *American Society of Animal Sciences*. DOI: 10.2527/jas2012-6162
- R Core Team. 2015. A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
- Sheather, S.J. 2008. *A Modern Approach to Regression with R*. Springer Texts in Statistics Original edition. ISBN 9780387096087
- Swindell, G.S. 2012. Eagle Ford Shale - an Early Look at Ultimate Recovery. Society of Petroleum Engineers. DOI: 10.2118/158207-MS.
- Valko, P.P. and Lee, W.J. 2010. A Better Way to Forecast Production from Unconventional Gas Wells. Society of Petroleum Engineers. DOI: 10.2118/134231-MS.

APPENDIX A

FIGURES AND CODE

Figure 2

```
temp_data <- iris[which(iris$Species=="virginica"),]
xyplot(Sepal.Length~Petal.Length,
       data=temp_data,type = c("p","g","r"),
       border = TRUE, main = list(label="Sepal Length vs. Petal Length (Virginica)",cex=2.5),
       lines = TRUE,
       xlab=list(label="Petal Length",cex=1.5),ylab=list(label="Sepal Length",cex=1.5),
       auto.key = list(title =
           "Legend",space="bottom",columns=3,border=TRUE,background="white" ),
       pch=16,col="red4",cex=1.5,scales=list(relation="same",tick.number=12,cex=1.3),
       xscale.components = xscale.components.1,yscale.components=yscale.components.1)
```

Figure 10

```
histogram(lastrate_data$Latest.Liquid,breaks=600,border = TRUE, main = list(label="Histogram of Latest
Liquid Rate",cex=2.5),
lines = TRUE,xlab=list(label="Latest Liquid Rate
(STB/Month)",cex=2.5),xlim=c(0,3000),ylab=list(label="Relative Frequency (%)",cex=2.5),
auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),
col="red4",scales=list(relation="same",tick.number=12,cex=1.3),
xscale.components =
xscale.components.subticks,yscale.components=yscale.components.subticks)
```

Figure 12

```
par(cex.axis=1,mar=c(8,4, 1, 1) + 0.1,las=2)
boxplot(fulldata$Calculated.EUR~fulldata$County.Name,plot=TRUE, na.rm=TRUE,
       col = (c("red4", "grey")),range=0,
       ylim=c(0,900000),outline = FALSE,medcol="white",medlwd=1)
mtext("County",side=1,line=7,las=0,cex=1.3)
mtext("Boxplot of EUR for Counties",side=3,line=0,las=0,cex=1.4)
means <- tapply(fulldata$Calculated.EUR,fulldata$County.Name,mean,na.rm = TRUE)
points(means,col="black",pch=7)
```

Figure 13

```
par(cex.axis=1,mar=c(8, 4, 1, 1) + 0.1,las=2)
boxplot(temp_data$Calculated.EUR~temp_data$Operator,range=0,plot=TRUE, na.rm=TRUE,col =
        (c("red4","grey")),
        ylim=c(0,900000),medcol="white",medlwd=1,na.rm = TRUE)
mtext("Operator",side=1,line=7,las=0,cex=1.3)
mtext("Boxplot of EUR(STB) for Operators",side=3,line=0,las=0,cex=1.4)
means <- tapply(temp_data$Calculated.EUR,temp_data$Operator,mean,na.rm = TRUE)
points(means,col="black",pch=7)
```

Figure 16

```
#Scatter plot of EUR vs. Peak oil
#####
#Formatting the numbers in the x-axis with an addition of comma
xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
  format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

#Formatting the numbers in the y-axis with an addition of comma
yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
  format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

#Scatter plot of EUR vs. Peak oil

xyplot(Calculated.EUR~`Peak oil`,
        data=temp_data,type = c("p","g","r"),
        border = TRUE, main = list(label="EUR Vs. Peak Oil",cex=2.5),
        lines = TRUE,
        xlab=list(label="Peak Oil (STB/month)",cex=1.5),ylab=list(label="EUR (STB)",cex=1.5),
        auto.key = list(title = "Legend",x = 1, y=1.0, corner =
        c(1,1),columns=1,border=TRUE,background="white" ),
        pch=16,col="blue",cex=1.1,scales=list(relation="same",tick.number=12,cex=1.3),xscale.components
        = xscale.components.1,yscale.components=yscale.components.1)
```

Figure 17

```
#Scatter plot of EUR vs. Liquid Gravity
#####
#Formatting the numbers in the x-axis with an addition of comma
xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
  format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

#Formatting the numbers in the y-axis with an addition of comma
yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
  format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

#Scatter plot of EUR vs. liquid gravity
xyplot(Calculated.EUR~Liquid.Gravity,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. Liquid Gravity",cex=2.5),
  lines = TRUE,
  xlab=list(label="Liquid Gravity",cex=1.5),ylab=list(label="EUR (STB)",cex=1.5),
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
  c(1,1),columns=1,border=TRUE,background="white" ),
  pch=16,cex=1.1,scales=list(relation="same",tick.number=12,cex=1.3),xscale.components =
  xscale.components.1,yscale.components=yscale.components.1)
```

Figure 18

```
#Scatter plot of EUR vs. Total Depth
#####
#Formatting the numbers in the x-axis with an addition of comma
xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
  format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

#Formatting the numbers in the y-axis with an addition of comma
yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
  format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}
```

```

}

#Scatter plot of EUR vs. Total Depth
xyplot(Calculated.EUR~Total.Depth,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. Total Depth",cex=2.5),
  lines = TRUE,
  xlab=list(label="Total Depth (Ft.)",cex=1.5),ylab=list(label="EUR (STB)",cex=1.5),
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),
  pch=16,cex=1.1,scales=list(relation="same",tick.number=12,cex=1.3),xscale.components =
xscale.components.1,yscale.components=yscale.components.1)

```

Figure 19

#Formatting the numbers in the x-axis with an addition of comma

```

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

```

#Formatting the numbers in the y-axis with an addition of comma

```

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

```

#Scatter plot of EUR vs. First month's water production

```

xyplot(Calculated.EUR~First.Month.Wtr,
  data=fulldata_no_NA,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. First Month Water Production",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="First Month Water Production (Bbl)",cex=1.5),ylab=list(label="EUR
(STB)",cex=1.5),
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),
  pch=16,col="blue",cex=1.1,scales=list(relation="same",tick.number=12,cex=1.3),xscale.components
= xscale.components.1,yscale.components=yscale.components.1)

```

Figure 20

```
temp_data <- fulldata_no_NA[which(fulldata_no_NA$First.Month.Wtr < 4000),]

#Formatting the numbers in the x-axis with an addition of comma

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

#Formatting the numbers in the y-axis with an addition of comma

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

#Scatter plot of EUR vs. First month's water production

xyplot(Calculated.EUR~First.Month.Wtr,
  data=temp_data, type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. First Month Water Production",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="First Month Water Production (Bbl)",cex=1.5),ylab=list(label="EUR
(STB)",cex=1.5),
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),
  pch=16,col="blue",cex=1.1,scales=list(relation="same",tick.number=12,cex=1.3),xscale.components
= xscale.components.1,yscale.components=yscale.components.1)
```

Figure 21

```
#Scatter plot of EUR vs. First month's water production for Atascosa

temp_data <- fulldata_no_NA [which(fulldata_no_NA$County.Name=="ATASCOSA (TX)",)]

#Formatting the numbers in the x-axis with an addition of comma

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}
```

#Formatting the numbers in the y-axis with an addition of comma

```
yscale.components.1 = function(...) {  
  yc <- yscale.components.default(...)  
  yc$left$labels$labels <-  
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")  
  yc  
}
```

#Scatter plot of EUR vs. First month's water production for Atascosa

```
xyplot(Calculated.EUR~First.Month.Wtr,  
  data=temp_data,type = c("p","g","r"),  
  border = TRUE, main = list(label="EUR Vs. First Month Water Production (ATASCOSA)",cex=2.5),  
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),  
  xlab=list(label="First Month Water Production (Bbl)",cex=1.5),ylab=NULL,  
  auto.key = list(title = "Legend",space="bottom",columns=3,border=TRUE,background="white" ),  
  pch=16,col="red4",cex=1.5,scales=list(relation="same",tick.number=12,cex=2),xscale.components =  
  xscale.components.1,yscale.components=yscale.components.1)
```

```
temp_data <- fulldata_no_NA [which(fulldata_no_NA$County.Name=="WILSON (TX)),]
```

#Formatting the numbers in the x-axis with an addition of comma

```
xscale.components.1 = function(...) {  
  xc <- xscale.components.default(...)  
  xc$bottom$labels$labels <-  
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")  
  xc  
}
```

#Formatting the numbers in the y-axis with an addition of comma

```
yscale.components.1 = function(...) {  
  yc <- yscale.components.default(...)  
  yc$left$labels$labels <-  
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")  
  yc  
}
```

#Scatter plot of EUR vs. First month's water production for Wilson

```
xyplot(Calculated.EUR~First.Month.Wtr,  
  data=temp_data,type = c("p","g","r"),  
  border = TRUE, main = list(label="EUR Vs. First Month Water Production (WILSON)",cex=2.5),  
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),  
  xlab=list(label="First Month Water Production (Bbl)",cex=1.5),ylim=c(0,320000),ylab=NULL,  
  auto.key = list(title = "Legend",space="bottom",columns=3,border=TRUE,background="white" ),  
  pch=16,col="red4",cex=1.5,scales=list(relation="same",tick.number=12,cex=2),xscale.components =  
  xscale.components.1,yscale.components=yscale.components.1)
```



```

temp_data <- fulldata_no_NA [which(fulldata_no_NA$County.Name=="LASALLE (TX)),]

#Formatting the numbers in the x-axis with an addition of comma

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

#Formatting the numbers in the y-axis with an addition of comma

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

#Scatter plot of EUR vs. First month's water production for Lasalle

xyplot(Calculated.EUR~First.Month.Wtr,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. First Month Water Production (LASALLE)",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="First Month Water Production (Bbl)",cex=1.5),ylim=c(0,320000),ylab=NULL,
  auto.key = list(title = "Legend",space="bottom",columns=3,border=TRUE,background="white" ),
  pch=16,col="red4",cex=1.5,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
xscale.components.1,yscale.components=yscale.components.1)

temp_data <- fulldata_no_NA [which(fulldata_no_NA$County.Name=="GONZALES (TX)),]

#Formatting the numbers in the x-axis with an addition of comma

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

#Formatting the numbers in the y-axis with an addition of comma

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

```

```
#Scatter plot of EUR vs. First month's water production for Gonzales
```

```
xyplot(Calculated.EUR~First.Month.Wtr,  
       data=temp_data,type = c("p","g","r"),  
       border = TRUE, main = list(label="EUR Vs. First Month Water Production (GONZALES)",cex=2.5),  
       lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),  
       xlab=list(label="First Month Water Production (Bbl)",cex=1.5),ylim=c(0,320000),ylab=NULL,  
       auto.key = list(title = "Legend",space="bottom",columns=3,border=TRUE,background="white" ),  
       pch=16,col="red4",cex=1.5,scales=list(relation="same",tick.number=12,cex=2),xscale.components =  
       xscale.components.1,yscale.components=yscale.components.1)
```

Figure 22

```
#Scatter plot of EUR vs. First month's water production for EOG
```

```
temp_data <- fulldata_no_NA[which(fulldata_no_NA$Operator=="EOG RESOURCES, INC."),]
```

```
#Formatting the numbers in the x-axis with an addition of comma
```

```
xscale.components.1 = function(...) {  
  xc <- xscale.components.default(...)  
  xc$bottom$labels$labels <-  
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")  
  xc  
}
```

```
#Formatting the numbers in the y-axis with an addition of comma
```

```
yscale.components.1 = function(...) {  
  yc <- yscale.components.default(...)  
  yc$left$labels$labels <-  
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")  
  yc  
}
```

```
#Scatter plot of EUR vs. First month's water production for EOG
```

```
xyplot(Calculated.EUR~First.Month.Wtr,  
       data=temp_data,type = c("p","g","r"),  
       border = TRUE, main = list(label="EUR Vs. First Month Water Production (EOG)",cex=2.5),  
       lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),  
       xlab=list(label="First Month Water Production (Bbl)",cex=1.5),ylab=NULL,  
       auto.key = list(title = "Legend",space="bottom",columns=3,border=TRUE,background="white" ),  
       pch=16,col="red4",cex=1.5,scales=list(relation="same",tick.number=12,cex=2),xscale.components =  
       xscale.components.1,yscale.components=yscale.components.1)
```

```
temp_data <- fulldata_no_NA[which(fulldata_no_NA$Operator=="CHESAPEAKE OPERATING,  
LLC"),]
```

```
#Scatter plot of EUR vs. First month's water production for Chesapeake
```

```
#Formatting the numbers in the x-axis with an addition of comma
```

```
xscale.components.1 = function(...) {  
  xc <- xscale.components.default(...)  
  xc$bottom$labels$labels <-  
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")  
  xc  
}
```

```
#Formatting the numbers in the y-axis with an addition of comma
```

```
yscale.components.1 = function(...) {  
  yc <- yscale.components.default(...)  
  yc$left$labels$labels <-  
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")  
  yc  
}
```

```
#Scatter plot of EUR vs. First month's water production for Chesapeake
```

```
xyplot(Calculated.EUR~First.Month.Wtr,  
  data=temp_data,type = c("p","g","r"),  
  border = TRUE, main = list(label="EUR Vs. First Month Water Production  
(CHESAPEAKE)",cex=2.5),  
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),  
  xlab=list(label="First Month Water Production (Bbl)",cex=1.5),ylab=NULL,  
  auto.key = list(title = "Legend",space="bottom",columns=3,border=TRUE,background="white" ),  
  pch=16,col="red4",cex=1.5,scales=list(relation="same",tick.number=12,cex=2),xscale.components =  
  xscale.components.1,yscale.components=yscale.components.1)
```

```
temp_data <- fulldata_no_NA[which(fulldata_no_NA$Operator=="MARATHON OIL EF LLC"),]
```

```
#Scatter plot of EUR vs. First month's water production for Marathon
```

```
#Formatting the numbers in the x-axis with an addition of comma
```

```
xscale.components.1 = function(...) {  
  xc <- xscale.components.default(...)  
  xc$bottom$labels$labels <-  
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")  
  xc  
}
```

```
#Formatting the numbers in the y-axis with an addition of comma
```

```
yscale.components.1 = function(...) {  
  yc <- yscale.components.default(...)  
  yc$left$labels$labels <-
```

```

    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  }
}

#Scatter plot of EUR vs. First month's water production for Marathon

xyplot(Calculated.EUR~First.Month.Wtr,
       data=temp_data,type = c("p","g","r"),
       border = TRUE, main = list(label="EUR Vs. First Month Water Production
(MARATHON)",cex=2.5),
       lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
       xlab=list(label="First Month Water Production (Bbl)",cex=1.5),ylab=NULL,
       auto.key = list(title = "Legend",space="bottom",columns=3,border=TRUE,background="white" ),
       pch=16,col="red4",cex=1.5,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
xscale.components.1,yscale.components=yscale.components.1)

temp_data <- fulldata_no_NA[which(fulldata_no_NA$Operator=="EP ENERGY E&P COMPANY,
LP"),]

#Scatter plot of EUR vs. First month's water production for EP ENERGY

#Formatting the numbers in the x-axis with an addition of comma

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

#Formatting the numbers in the y-axis with an addition of comma

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

#Scatter plot of EUR vs. First month's water production for EP Energy

xyplot(Calculated.EUR~First.Month.Wtr,
       data=temp_data,type = c("p","g","r"),
       border = TRUE, main = list(label="EUR Vs. First Month Water Production (EP
ENERGY)",cex=2.5),
       lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
       xlab=list(label="First Month Water Production (Bbl)",cex=1.5),ylab=NULL,
       auto.key = list(title = "Legend",space="bottom",columns=3,border=TRUE,background="white" ),
       pch=16,col="red4",cex=1.5,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
xscale.components.1,yscale.components=yscale.components.1)

```

Figure 23

```
#Scatter plot of EUR vs. Lateral Length

#Formatting the numbers in the x-axis with an addition of comma

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

#Formatting the numbers in the y-axis with an addition of comma

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

xyplot(Calculated.EUR~Lateral.Length,
  data=fulldata_no_NA,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. Lateral Length",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="Lateral Length (Ft.)",cex=1.5),ylab= list(label="EUR (STB)",cex=1.5),
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),
  pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
xscale.components.1,yscale.components=yscale.components.1)
```

Figure 24

```
temp_data <- fulldata_no_NA[which(fulldata_no_NA$County.Name=="ATASCOSA (TX)),]

#Scatter plot of EUR vs. Lateral Length for Atascosa

#Formatting the numbers in the x-axis with an addition of comma

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}
```

```

}

#Formatting the numbers in the y-axis with an addition of comma

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

#Scatter plot of EUR vs. LL for Atascosa

xyplot(Calculated.EUR~Lateral.Length,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. Lateral Length (ATASCOSA)",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="Lateral Length (Ft.)",cex=1.5),ylab=NULL,ylim=c(0,340000),
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),
  pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
xscale.components.1,yscale.components=yscale.components.1)

temp_data <- fulldata_no_NA[which(fulldata_no_NA$County.Name=="DEWITT (TX)),]

#Formatting the numbers in the x-axis with an addition of comma

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

#Formatting the numbers in the y-axis with an addition of comma

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

#Scatter plot of EUR vs. LL for Dewitt

xyplot(Calculated.EUR~Lateral.Length,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. Lateral Length (DEWITT)",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="Lateral Length (Ft.)",cex=1.5),ylab=NULL,ylim=c(0,340000),
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),

```

```
pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
yscale.components.1,yscale.components=yscale.components.1)
```

```
temp_data <- fulldata_no_NA[which(fulldata_no_NA$County.Name=="GONZALES (TX)),]
```

```
#Formatting the numbers in the x-axis with an addition of comma
```

```
xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}
```

```
#Formatting the numbers in the y-axis with an addition of comma
```

```
yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}
```

```
#Scatter plot of EUR vs. LL for Gonzales
```

```
xyplot(Calculated.EUR~Lateral.Length,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. Lateral Length (GONZALES)",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="Lateral Length (Ft.)",cex=1.5),ylab=NULL,ylim=c(0,340000),
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),
  pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
yscale.components.1,yscale.components=yscale.components.1)
```

```
temp_data <- fulldata_no_NA[which(fulldata_no_NA$County.Name=="KARNES (TX)),]
```

```
#Formatting the numbers in the x-axis with an addition of comma
```

```
xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}
```

```
#Formatting the numbers in the y-axis with an addition of comma
```

```
yscale.components.1 = function(...) {
```

```

yc <- yscale.components.default(...)
yc$left$labels$labels <-
  format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
yc
}

#Scatter plot of EUR vs. LL for Karnes

xyplot(Calculated.EUR~Lateral.Length,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. Lateral Length (KARNES)",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="Lateral Length (Ft.)",cex=1.5),ylab=NULL,ylim=c(0,340000),
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),
  pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
xscale.components.1,yscale.components=yscale.components.1)

#Scatter plot of single operator to understand the behaviour of EUR vs Lateral Length

temp_data <- fulldata_no_NA[which(fulldata_no_NA$Operator=="EOG RESOURCES, INC."),]

#Scatter plot of EUR vs. Lateral Length for EOG

#Formatting the numbers in the x-axis with an addition of comma

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

#Formatting the numbers in the y-axis with an addition of comma

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

#Scatter plot of EUR vs. First month's water production for EOG

xyplot(Calculated.EUR~Lateral.Length,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. Lateral Length (EOG)",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="Lateral Length (Ft.)",cex=1.5),ylab=NULL,ylim = c(0,320000),
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),

```



```
pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
yscale.components.1,yscale.components=yscale.components.1)
```

```
temp_data <- fulldata_no_NA[which(fulldata_no_NA$Operator=="CHESAPEAKE OPERATING,
LLC"),]
```

```
#Formatting the numbers in the x-axis with an addition of comma
```

```
xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}
```

```
#Formatting the numbers in the y-axis with an addition of comma
```

```
yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}
```

```
#Scatter plot of EUR vs. Lateral Length for Chesapeake
```

```
xyplot(Calculated.EUR~Lateral.Length,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. Lateral Length (CHESAPEAKE)",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="Lateral Length (Ft.)",cex=1.5),ylab=NULL,ylim=c(0,320000),
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),
  pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
yscale.components.1,yscale.components=yscale.components.1)
```

```
temp_data <- fulldata_no_NA[which(fulldata_no_NA$Operator=="MARATHON OIL EF LLC"),]
```

```
#Scatter plot of EUR vs. Lateral Length for Marathon
```

```
#Formatting the numbers in the x-axis with an addition of comma
```

```
xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}
```

```
#Formatting the numbers in the y-axis with an addition of comma
```

```

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

```

#Scatter plot of EUR vs. Lateral Length for Marathon

```

xyplot(Calculated.EUR~Lateral.Length,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. Lateral Length (MARATHON)",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="Lateral Length (Ft.)",cex=1.5),ylab=NULL, ylim=c(0,320000),
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),
  pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
yscale.components.1,yscale.components=yscale.components.1)

```

```

temp_data <- fulldata_no_NA[which(fulldata_no_NA$Operator=="EP ENERGY E&P COMPANY,
LP"),]

```

#Scatter plot of EUR vs. Lateral.Length for EP ENERGY

#Formatting the numbers in the x-axis with an addition of comma

```

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

```

#Formatting the numbers in the y-axis with an addition of comma

```

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

```

#Scatter plot of EUR vs. Lateral.Length for EP ENERGY

```

xyplot(Calculated.EUR~Lateral.Length,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. Lateral Length (EP ENERGY)",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="Lateral Length (Ft.)",cex=1.5),ylab=NULL, ylim=c(0,320000),

```

```

auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),
pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
xscale.components.1,yscale.components=yscale.components.1)

```

Figure 25

#Scatter plot of EUR vs. GOR

```
temp_data <- fulldata_no_NA[which(fulldata_no_NA$GOR < 20000),]
```

#Formatting the numbers in the x-axis with an addition of comma

```

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

```

#Formatting the numbers in the y-axis with an addition of comma

```

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

```

#Scatter plot of EUR vs. GOR

```

xyplot(Calculated.EUR~GOR,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. GOR",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="GOR (Mscf/STB)",cex=1.5),ylab=list(label="EUR (STB)",cex=1.5),
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),
  pch=16,col="blue",cex=1.1,scales=list(relation="same",tick.number=12,cex=1.3),xscale.components
= xscale.components.1,yscale.components=yscale.components.1)

```

#Scatter plot of EUR vs. GOR

```
temp_data <- fulldata_no_NA[which(fulldata_no_NA$GOR < 10000),]
```

#Formatting the numbers in the x-axis with an addition of comma

```

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)

```

```

xc$bottom$labels$labels <-
  format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
xc
}

#Formatting the numbers in the y-axis with an addition of comma

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

#Scatter plot of EUR vs. GOR

xyplot(Calculated.EUR~GOR,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. GOR",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="GOR (Mscf/STB)",cex=1.5),ylab=list(label="EUR (STB)",cex=1.5),
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),
  pch=16,col="blue",cex=1.1,scales=list(relation="same",tick.number=12,cex=1.3),xscale.components
= xscale.components.1,yscale.components=yscale.components.1)

```

Figure 26

```

#Scatter plots of EUR vs GOR for different counties and operators

temp_data <- fulldata_no_NA[which(fulldata_no_NA$County.Name=="ATASCOSA (TX)",)]

#Formatting the numbers in the x-axis with an addition of comma

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

#Formatting the numbers in the y-axis with an addition of comma

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

```

```
#Scatter plot of EUR vs. GOR
```

```
xyplot(Calculated.EUR~GOR,  
  data=temp_data,type = c("p","g","r"),  
  border = TRUE, main = list(label="EUR Vs. GOR (ATASCOSA)",cex=2.5),  
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),  
  xlab=list(label="GOR (Mscf/STB)",cex=1.5),ylab=NULL,  
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =  
  c(1,1),columns=1,border=TRUE,background="white" ),  
  pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =  
  xscale.components.1,yscale.components=yscale.components.1)
```

```
temp_data <- fulldata_no_NA[which(fulldata_no_NA$County.Name=="DIMMIT (TX)),]
```

```
#Formatting the numbers in the x-axis with an addition of comma
```

```
xscale.components.1 = function(...) {  
  xc <- xscale.components.default(...)  
  xc$bottom$labels$labels <-  
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")  
  xc  
}
```

```
#Formatting the numbers in the y-axis with an addition of comma
```

```
yscale.components.1 = function(...) {  
  yc <- yscale.components.default(...)  
  yc$left$labels$labels <-  
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")  
  yc  
}
```

```
#Scatter plot of EUR vs. GOR
```

```
xyplot(Calculated.EUR~GOR,  
  data=temp_data,type = c("p","g","r"),  
  border = TRUE, main = list(label="EUR Vs. GOR (DIMMIT)",cex=2.5),  
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),  
  xlab=list(label="GOR (Mscf/STB)",cex=1.5),ylab=NULL,  
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =  
  c(1,1),columns=1,border=TRUE,background="white" ),  
  pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =  
  xscale.components.1,yscale.components=yscale.components.1)
```

```
temp_data <- fulldata_no_NA[which(fulldata_no_NA$County.Name=="GONZALES (TX)),]
```

```
#Formatting the numbers in the x-axis with an addition of comma
```

```
xscale.components.1 = function(...) {  
  xc <- xscale.components.default(...)
```

```

xc$bottom$labels$labels <-
  format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
xc
}

#Formatting the numbers in the y-axis with an addition of comma

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

#Scatter plot of EUR vs. GOR

xyplot(Calculated.EUR~GOR,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. GOR (GONZALES)",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="GOR (Mscf/STB)",cex=1.5),ylab=NULL,
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
  c(1,1),columns=1,border=TRUE,background="white" ),
  pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
  xscale.components.1,yscale.components=yscale.components.1)

temp_data <- fulldata_no_NA[which(fulldata_no_NA$County.Name=="LIVE OAK (TX)",)]

#Formatting the numbers in the x-axis with an addition of comma

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

#Formatting the numbers in the y-axis with an addition of comma

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

#Scatter plot of EUR vs. GOR

xyplot(Calculated.EUR~GOR,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. GOR (LIVE OAK)",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),

```

```

xlab=list(label="GOR (Mscf/STB)",cex=1.5),ylab=NULL,
auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),
pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
xscale.components.1,yscale.components=yscale.components.1)

#Scatter plot of EUR vs GOR for operators

temp_data <- fulldata_no_NA[which(fulldata_no_NA$Operator=="EOG RESOURCES, INC."),]

#Formatting the numbers in the x-axis with an addition of comma

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

#Formatting the numbers in the y-axis with an addition of comma

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

#Scatter plot of EUR vs. GOR

xyplot(Calculated.EUR~GOR,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. GOR (EOG)",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="GOR (Mscf/STB)",cex=1.5),ylab=NULL,
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),
  pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
xscale.components.1,yscale.components=yscale.components.1)

temp_data <- fulldata_no_NA[which(fulldata_no_NA$Operator=="CHESAPEAKE OPERATING,
  LLC"),]

#Formatting the numbers in the x-axis with an addition of comma

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

```

#Formatting the numbers in the y-axis with an addition of comma

```
yscale.components.1 = function(...) {  
  yc <- yscale.components.default(...)  
  yc$left$labels$labels <-  
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")  
  yc  
}
```

#Scatter plot of EUR vs. GOR

```
xyplot(Calculated.EUR~GOR,  
  data=temp_data,type = c("p","g","r"),  
  border = TRUE, main = list(label="EUR Vs. GOR (CHESAPEAKE)",cex=2.5),  
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),  
  xlab=list(label="GOR (Mscf/STB)",cex=1.5),ylab=NULL,  
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =  
    c(1,1),columns=1,border=TRUE,background="white" ),  
  pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =  
  xscale.components.1,yscale.components=yscale.components.1)
```

```
temp_data <- fulldata_no_NA[which(fulldata_no_NA$Operator=="BURLINGTON RESOURCES OIL &  
  GAS CO"),]
```

#Formatting the numbers in the x-axis with an addition of comma

```
xscale.components.1 = function(...) {  
  xc <- xscale.components.default(...)  
  xc$bottom$labels$labels <-  
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")  
  xc  
}
```

#Formatting the numbers in the y-axis with an addition of comma

```
yscale.components.1 = function(...) {  
  yc <- yscale.components.default(...)  
  yc$left$labels$labels <-  
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")  
  yc  
}
```

#Scatter plot of EUR vs. GOR

```
xyplot(Calculated.EUR~GOR,  
  data=temp_data,type = c("p","g","r"),  
  border = TRUE, main = list(label="EUR Vs. GOR (BURLINGTON)",cex=2.5),  
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),  
  xlab=list(label="GOR (Mscf/STB)",cex=1.5),ylab=NULL,  
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =  
    c(1,1),columns=1,border=TRUE,background="white" ),
```



```

pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
xscale.components.1,yscale.components=yscale.components.1)

temp_data <- fulldata_no_NA[which(fulldata_no_NA$Operator=="MARATHON OIL EF LLC"),]

#Formatting the numbers in the x-axis with an addition of comma

xscale.components.1 = function(...) {
  xc <- xscale.components.default(...)
  xc$bottom$labels$labels <-
    format(xc$bottom$labels$at, scientific = FALSE, big.mark = ",")
  xc
}

#Formatting the numbers in the y-axis with an addition of comma

yscale.components.1 = function(...) {
  yc <- yscale.components.default(...)
  yc$left$labels$labels <-
    format(yc$left$labels$at, scientific = FALSE, big.mark = ",")
  yc
}

#Scatter plot of EUR vs. GOR

xyplot(Calculated.EUR~GOR,
  data=temp_data,type = c("p","g","r"),
  border = TRUE, main = list(label="EUR Vs. GOR (MARATHON)",cex=2.5),
  lines = TRUE,strip = strip.custom(strip.names = TRUE,var.name = "Operator"),
  xlab=list(label="GOR (Mscf/STB)",cex=1.5),ylab=NULL,
  auto.key = list(title = "Legend",x = 1, y=1.0, corner =
c(1,1),columns=1,border=TRUE,background="white" ),
  pch=16,col="red4",cex=1.1,scales=list(relation="same",tick.number=12,cex=2),xscale.components =
xscale.components.1,yscale.components=yscale.components.1)

```

Figure 27

```

#Boxplot of EUR with outliers

par(cex.axis=2,mar=c(3, 6, 3, 3) + 0.1,las=2)
outliers<-boxplot(fulldata_no_NA$Calculated.EUR,plot=TRUE, na.rm=TRUE,
  col = "red4",
ylim=c(0,850000),medcol="white",medlwd=1,outcol="black",outpch=16)$out;outliers
mtext(" All data",side=1,line=1,las=0,cex=2.5)
mtext("Boxplot showing outliers (EUR)",side=3,line=0,las=0,cex=2.5)
points(mean(fulldata_no_NA$Calculated.EUR),col="white",pch=16,cex=2.5)

```

#Boxplot of EUR without outliers

```
par(cex.axis=2,mar=c(3,6,3,3) + 0.1,las=2)
outliers<-boxplot(fulldata_no_NA_no_outliers4$Calculated.EUR,plot=TRUE, na.rm=TRUE,
  col = "chartreuse4",
  ylim=c(0,850000),medcol="white",medlwd=1,outcol="black",outpch=16)$out; outliers
mtext("All data",side=1,line=1,las=0,cex=2.5)
#mtext("Estimated Ultimate Recovery (STB)",side=2,line=4,las=0,cex=2)
mtext("Boxplot of EUR data",side=3,line=0,las=0,cex=2.5)
points(mean(fulldata_no_NA_no_outliers4$Calculated.EUR),col="white",pch=16,cex=2.5)
```

#Boxplot of First month water production with outliers

```
par(cex.axis=2,mar=c(3, 6, 3, 3) + 0.1,las=2)
outliers<-boxplot(fulldata_no_NA$First.Month.Wtr,plot=TRUE, na.rm=TRUE,
  col = "red4", ylim=c(0,40000),medcol="white",medlwd=1,outcol="black",outpch=16)$out;
  outliers
mtext("All data",side=1,line=1,las=0,cex=2.5)
mtext("Boxplot showing outliers (First Month water production)",side=3,line=0,las=0,cex=2)
points(mean(fulldata_no_NA$First.Month.Wtr),col="white",pch=16,cex=2.5)
```

#Boxplot of First month water production without outliers

```
par(cex.axis=2,mar=c(3, 6, 3, 3) + 0.1,las=2)
outliers<-boxplot(fulldata_no_NA_no_outliers4$First.Month.Wtr,plot=TRUE, na.rm=TRUE,
  col = "chartreuse4",
  ylim=c(0,20000),medcol="white",medlwd=1,outcol="black",outpch=16)$out; outliers
mtext("All data",side=1,line=1,las=0,cex=2.5)
mtext("Boxplot (First month water production)",side=3,line=0,las=0,cex=2.5)
points(mean(fulldata_no_NA_no_outliers4$First.Month.Wtr),col="white",pch=16,cex=2.5)
```

#Boxplot of Lateral Length with outliers

```
par(cex.axis=2,mar=c(3, 6, 3, 3) + 0.1,las=2)
outliers<-boxplot(fulldata_no_NA$Lateral.Length,plot=TRUE, na.rm=TRUE,
  col = "red4", ylim=c(0,15000),medcol="white",medlwd=1,outcol="black",outpch=16)$out;
  outliers
mtext("All data",side=1,line=1,las=0,cex=2.5)
mtext("Boxplot showing outliers (Lateral Length)",side=3,line=0,las=0,cex=2.5)
points(mean(fulldata_no_NA$Lateral.Length),col="white",pch=16,cex=2.5)
```

#Boxplot of Lateral Length without outliers

```
par(cex.axis=2,mar=c(3, 6, 3, 3) + 0.1,las=2)
outliers<-boxplot(fulldata_no_NA_no_outliers4$Lateral.Length,plot=TRUE, na.rm=TRUE,
  col = "chartreuse4",
  ylim=c(1000,7000),medcol="white",medlwd=1,outcol="black",outpch=16)$out; outliers
mtext("All data",side=1,line=1,las=0,cex=2.5)
```

```
mtext("Boxplot (Lateral Length)",side=3,line=0,las=0,cex=2.5)
points(mean(fulldata_no_NA_no_outliers4$Lateral.Length),col="white",pch=16,cex=2.5)
```

#Boxplot of Peak oil with outliers

```
par(cex.axis=2,mar=c(3, 6, 3, 3) + 0.1,las=2)
outliers<-boxplot(fulldata_no_NA$`Peak oil`,plot=TRUE, na.rm=TRUE,
  col = "red4",
  ylim=c(0,55000),medcol="white",medlwd=1,outcol="black",outpch=16)$out;outliers
mtext("All data",side=1,line=1,las=0,cex=2.5)
mtext("Boxplot showing outliers (Peak Oil (STB/Month))",side=3,line=0,las=0,cex=2)
points(mean(fulldata_no_NA$`Peak oil`),col="white",pch=16,cex=2.5)
```

#Boxplot of Peak oil without outliers

```
par(cex.axis=2,mar=c(3, 6, 3, 3) + 0.1,las=2)
outliers<-boxplot(fulldata_no_NA_no_outliers4$Peak.oil,plot=TRUE, na.rm=TRUE,
  col = "chartreuse4",
  ylim=c(0,55000),medcol="white",medlwd=1,outcol="black",outpch=16)$out;outliers
mtext("All data",side=1,line=1,las=0,cex=2.5)
mtext("Boxplot (Peak Oil (STB/Month))",side=3,line=0,las=0,cex=2.5)
points(mean(fulldata_no_NA_no_outliers4$`Peak oil`),col="white",pch=16,cex=2.5)
```

Figure 28

#Boxplot of Liquid Gravity with outliers

```
par(cex.axis=2,mar=c(3, 6, 3, 3) + 0.1,las=2)
outliers<-boxplot(fulldata_no_NA_no_outliers4$Liquid.Gravity,plot=TRUE, na.rm=TRUE,
  col = "red4", ylim=c(0,70),medcol="white",medlwd=1,outcol="black",outpch=16)$out;
  outliers
mtext("All data",side=1,line=1,las=0,cex=2.5)
mtext("Boxplot showing outliers (Liquid Gravity)",side=3,line=0,las=0,cex=2.5)
points(mean(fulldata_no_NA_no_outliers4$Liquid.Gravity),col="white",pch=16,cex=2.5)
```

#No outlier is removed from Liquid Gravity

#Outlier removed in fulldata_no_NA_no_outliers4 are automatically from Liquid Gravity as well

#Boxplot of Total Depth with outliers

```
par(cex.axis=2,mar=c(3, 6, 3, 3) + 0.1,las=2)
outliers<-boxplot(fulldata_no_NA_no_outliers4$Total.Depth,plot=TRUE, na.rm=TRUE,
  col = "red4",
  ylim=c(5000,25000),medcol="white",medlwd=1,outcol="black",outpch=16)$out;outliers
mtext("All data",side=1,line=1,las=0,cex=2.5)
#mtext("Total Depth (Ft.)",side=2,line=4,las=0,cex=2.5)
mtext("Boxplot showing outliers (Total Depth)",side=3,line=0,las=0,cex=2.5)
points(mean(fulldata_no_NA_no_outliers4$Total.Depth),col="white",pch=16,cex=2.5)
```

```
#No outlier is removed from Total Depth
#Outlier removed in fulldata_no_NA_no_outliers4 are automatically from total depth as well
```

Figure 29

```
# Determine optimum number of clusters in the data

wss <- (nrow(clusterdata[,c(1:5)])-1)*sum(apply(clusterdata[,c(1:5)],2,var))
for (i in 2:20) wss[i] <- sum(kmeans(clusterdata[,c(1:5)],
                                centers=i)$withinss)
par(cex.axis=1,mar=c(5, 5, 3, 3) + 0.1,las=0)
plot(1:20, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares",main="Plot to find optimum number of clusters")
```

Figure 30

```
#3D Plot for the important input variables

plot3d(clusterdata$Total.Depth,clusterdata$First.Month.Wtr,clusterdata$Lateral.Length,col=as.integer(clusterdata$fitclus.cluster),xlab="Total Depth (Ft.)", ylab="First month water production (bbl)",zlab="Lateral Length (Ft.)")
```

Figure 35

```
#Histogram of relative frequency for absolute percentage errors for the testing dataset

histogram(perc_error_eur$`Error%`,type="percent",nint=21,equal.widths = TRUE,breaks=NULL,border = TRUE, main = list(label="Histogram of Errors",cex=2.5),
          lines = TRUE,xlab=list(label="Absolute Percentage Error (%)",cex=2.5),xlim=c(0,220),ylab=list(label="Relative Frequency (%)",cex=2.5),
          auto.key = list(title = "Legend",x = 1, y=1.0, corner = c(1,1),columns=1,border=TRUE,background="white" ),
          col="red4",scales=list(relation="same",tick.number=12,cex=1.3),
          xscale.components = xscale.components.subticks,yscale.components=yscale.components.subticks)
```

APPENDIX B

OUTLIER ANALYSIS

The data that has been used in this study may contain several outliers. Hence their analysis and removal involves a stepwise procedure. The procedure is followed for each of the parameters separately. The approach necessitates testing the data for outliers using boxplots. If these outliers removal improves the distribution of the data, the subsample remaining after their deletion is tested again for outliers. In this approach, we'll delete the outliers until their removal does not improve the normality of the distribution.

To explain the procedure, example of outlier removal from the parameter estimated ultimate recovery (EUR) is illustrated. The Fig. B-1 shows the boxplot with outliers as well as the quantile-quantile (q-q) plot for the raw data of EUR. The q-q plot shows whether the data is following a normal distribution or not. The data points falling inside the two dash red lines follow normal distribution. The data forms tails both at the beginning and the end which are not part of the normal distribution. Till now no outliers have been removed.

After the removal of these outliers, the q-q plot shows that normality of the data is improved significantly. We also observe new outliers in the boxplot. These plots are shown in Fig. B-2.

After another round of removal of outliers, no new outliers appear in the boxplot and there is a slight improvement in the normality of the EUR distribution (Fig. B-3). In this stepwise procedure outliers are removed from a parameter.

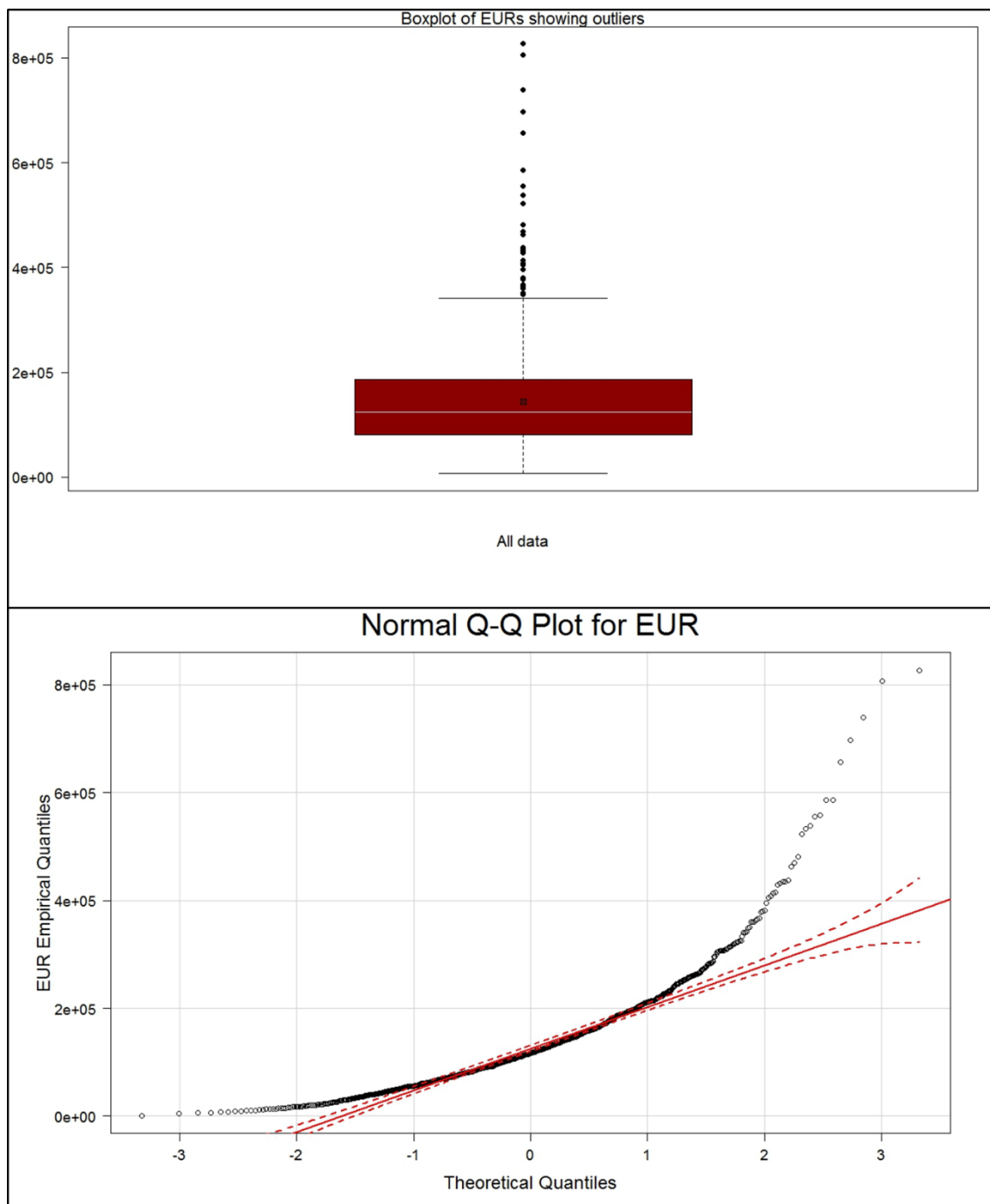


Figure B-1. The charts show the boxplot and the quantile-quantile plot for the raw EUR data. Note the outliers in the boxplot and the tails in the q-q plot.

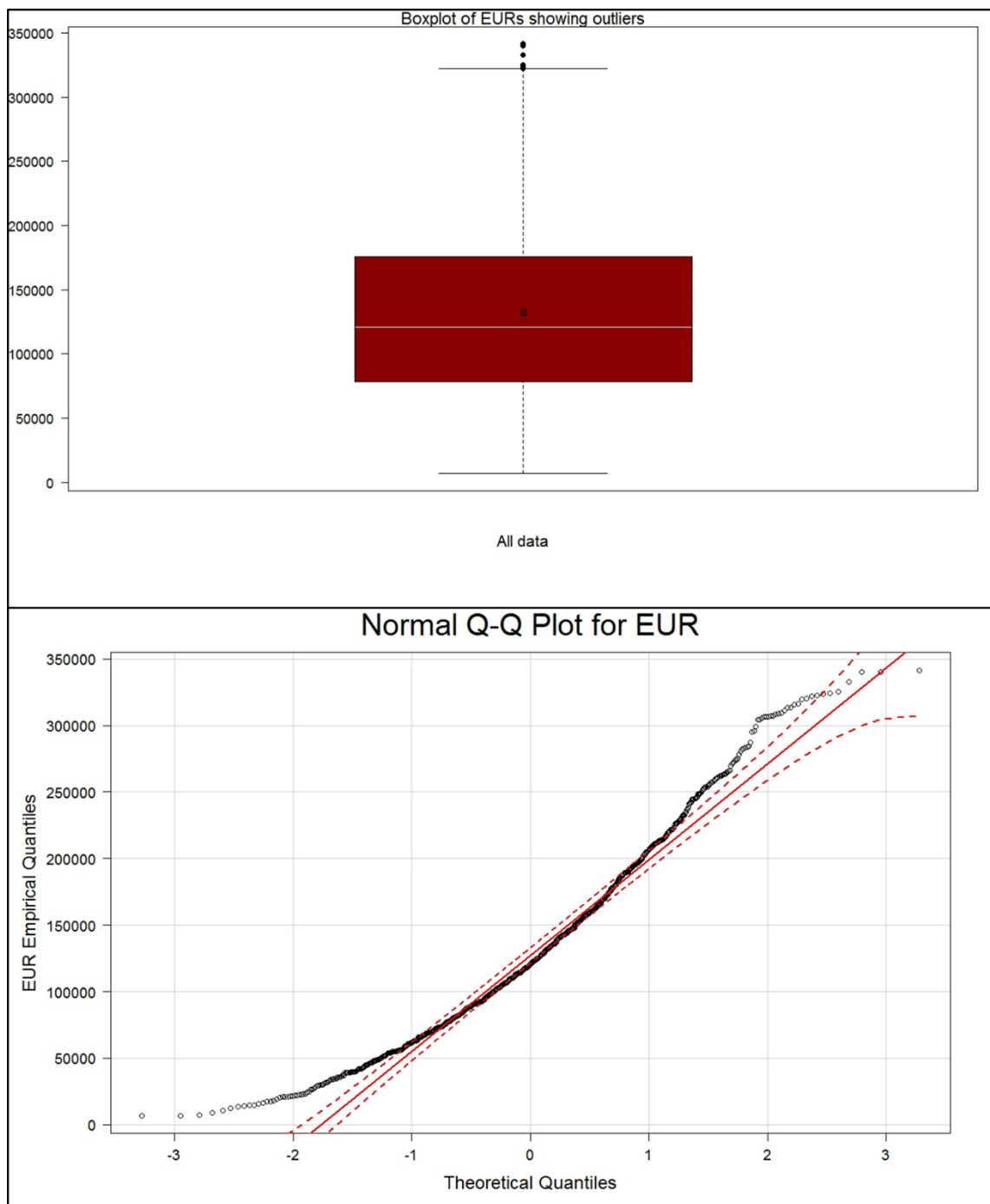


Figure B-2. The charts show the boxplot and the quantile-quantile plot for the EUR data after first round of removal of outliers. Note new outliers in the boxplot and the tails in the q-q plot.

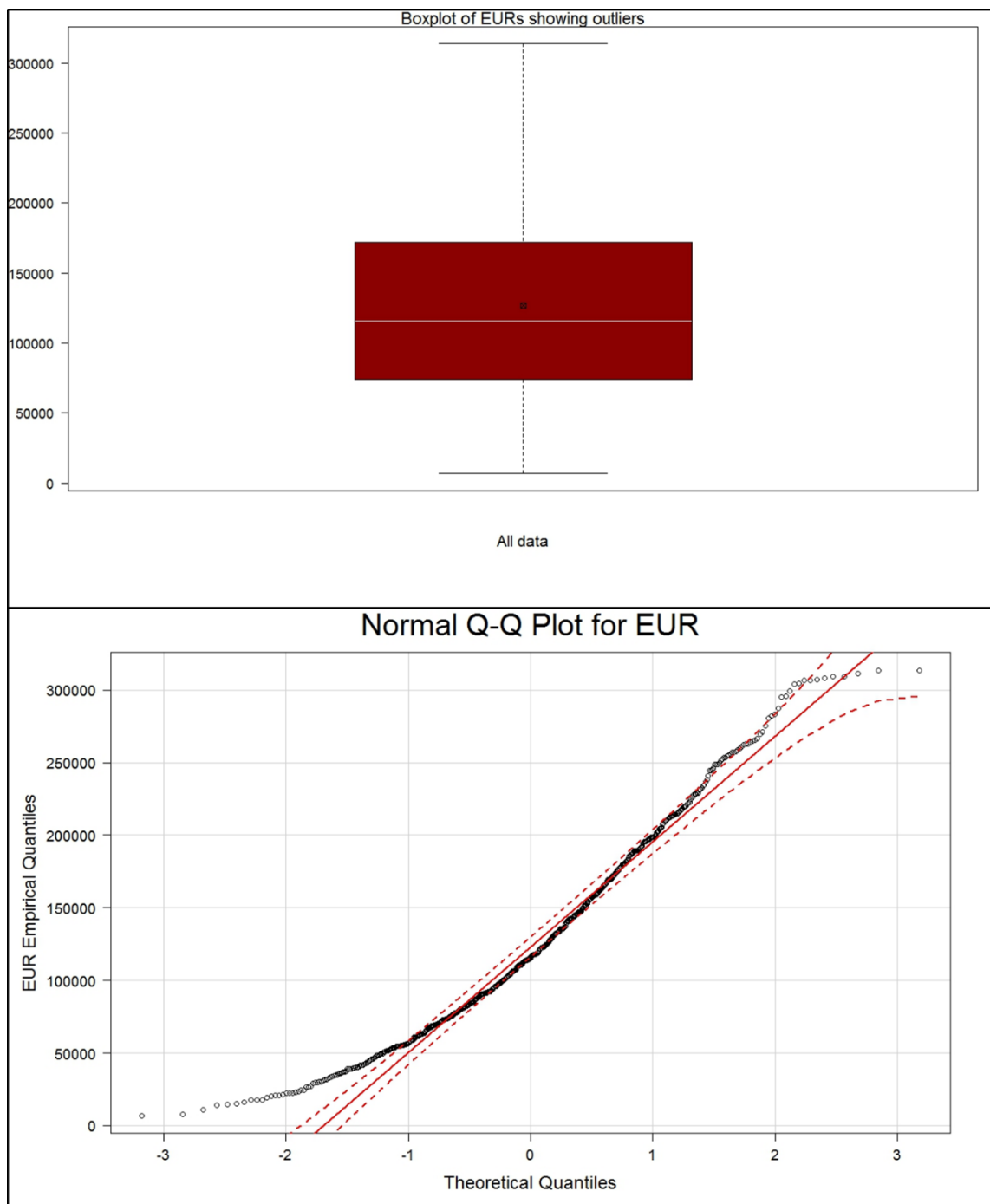


Figure B-3. The charts show the boxplot and the quantile-quantile plot for the EUR data after final removal of outliers.